1984

Speech code identification.

Vijay. Venkatachalam
University of Windsor

SPEECH CODE IDENTIFICATION

by

VIJAY VENKATACHALAM

A thesis
presented to the University of Windsor
in partial fulfillment of the
requirements for the degree of
MASTER OF APPLIED SCIENCE
in
Electrical Engineering

Windsor , Ontario, 1984

ABSTRACT

This thesis is an investigation into Code-Identification in a Digitally Coded Speech Signal using the techniques of Pattern Classification.

Features are selected ~~for this study by study~~ing the properties of the coded speech signals, and by specifying the information content they could provide, when extracted from the coded speech signal.

The Orthogonal Transformation and other normalization procedures are carried out on the above features to improve their code-discrimination properties. The effectiveness of these transformations and normalization procedures is tested using scatter plots.

Finally, a classification scheme for implementing the Code-Identification system is proposed, and the results obtained from it are discussed.

The results indicate that for accurate identification of the codes, a certain minimum duration of the coded signal is necessary. This minimum length of the signal is required for the purpose of averaging out the sentence or text-dependent properties of the transformed features which are derived from it, thus leaving predominantly the code-sensitive properties for the identification.

This minimum duration of the signal which is required is not the same for all the codes considered in this study, but depends on the code-set being considered for the identification. If only Pulse Amplitude Modulation, Log-PAM (U=100), First-Order Fixed Predictor DPAM and First Order Adaptive Predictor DPAM systems are considered for identification, then 16000 samples or 1.6 seconds of the coded signal is sufficient, but if Second Order Adaptive Predictor DPAM is included in the study then 8.0 seconds of the coded signal is necessary for accurate identification.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and thanks to Dr. M. Shridhar and Professor P. H. Alexander for the supervision, guidance and encouragement they bestowed upon me while carrying out this work.

I am indebted to them for going out of the way and helping me out when I was in the deepest of troubles.

I also thank Dr. Soltis and Dr. Ahmadi for their helpful comments and criticisms on this work.

I would also like to thank all the other faculty members and graduate students of the Electrical Engineering Department for their helpful attitudes and cooperation, which went a long way in helping me with my research.

I am also indebted to my parents and family members for their encouragement and loving patience, without which I could not have come this far.

Finally, I would like to thank Professor V. V. S. Sarma and Ms. Bharathi Devi, School of Automation, Indian Institute of Science, Bangalore, India, for introducing me to this fascinating field of Speech Processing, and for imbibing in me the concepts and techniques in this field, which have stood me in good stead while carrying out this work.

LIST OF ILLUSTRATIONS

CONTENTS

Chapter I

INTRODUCTION.

Digital signal processing is concerned with the obtaining of discrete representations of signals, and with the theory, design and implementation of numerical procedures for processing the discrete representations [1].

The advantages in using Digital Signal Processing in the area of Speech Processing are plenty.

First, extremely sophisticated signal processing functions can be implemented using digital techniques. Prior to the mid-1960's, essentially all speech processing systems were based on analog-hardware implementations, although a number of speech processing systems were implemented on general purpose computers. These systems however were viewed as non-real-time simulations of analog hardware systems, and were based on algorithms that were to match available analog hardware.

In recent years, more and more attention has been given to the use of Digital Signal Processing in speech communication systems. This is due in a large part to the increasing availability of medium and large scale digital integrated circuits with their desirable properties of small size, low power, low cost, noise immunity and reliability, and to the development of faster computers and new signal processing algorithms and techniques [2,4,5].

Second, suitable coding can enable speech in digital form to be reliably transmitted over very noisy channels.

Third, speech signals in digital form are identical to data signals. Thus a communications network can be used to transmit both speech and data signals, with no need to distinguish between them except in decoding.

Finally, a digital representation of speech signals has applications in communication systems where secrecy or security is of prime importance.

1.1 DIGITAL REPRESENTATION OF SPEECH SIGNALS

The digital representation of speech signals is guided by the well known Sampling Theorem, which states that a band-limited signal can be represented by samples taken periodically in time-provided that the sampling frequency is greater than twice the highest frequency component present in the signal. Fig.1.1 shows how the representation of speech can be broadly classified.

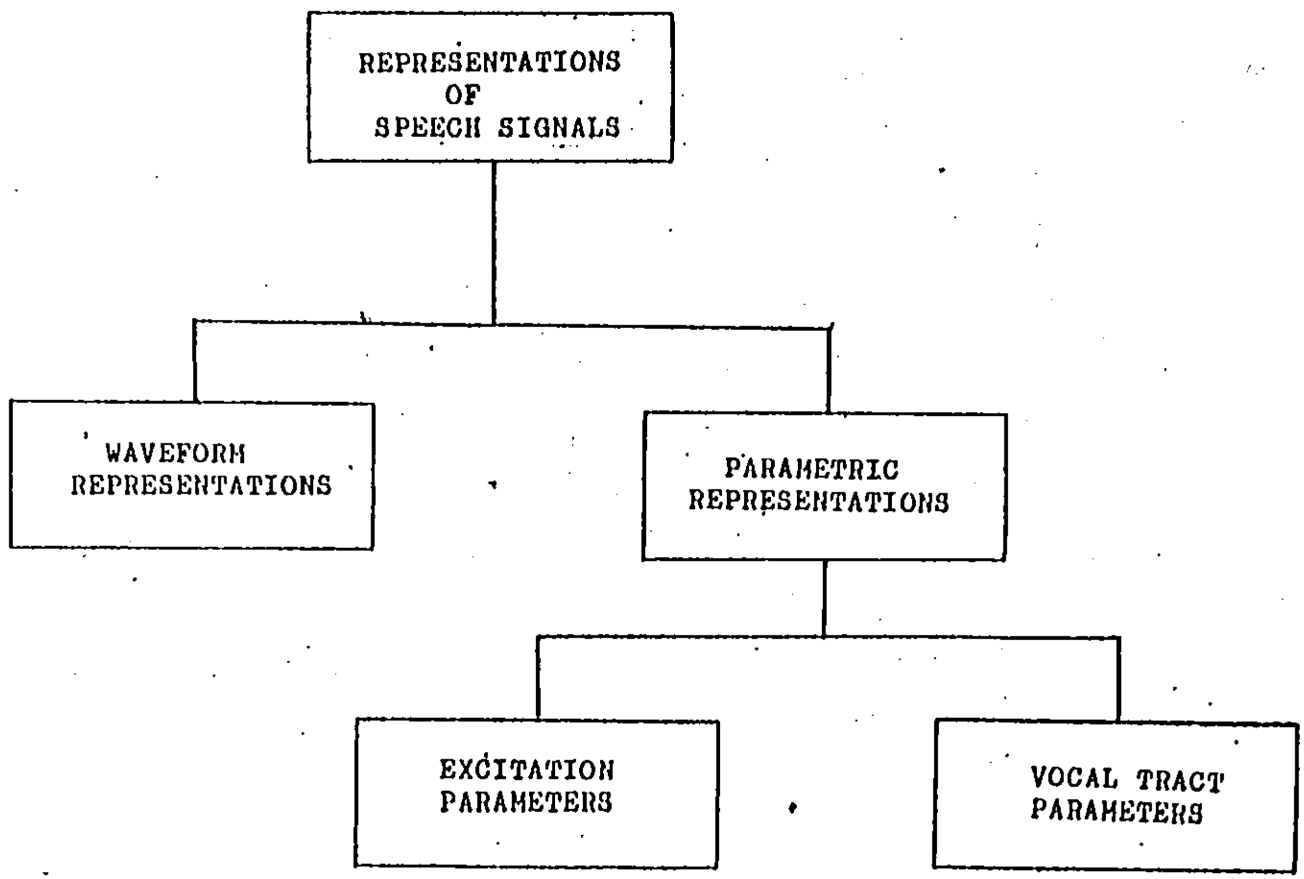


Fig.1.1. Digital Representations of Speech Signals

The two main groups are :-

i) Waveform Representations

ii) Parametric Representations

1.1.1 Waveform Representations

The Waveform Representation of speech signals is concerned with preserving the 'waveshape' of the analog speech signal through a sampling and a quantization process.

1.1.2 Parametric Representations

Parametric representations are concerned with representing the speech signal as the output of a model for speech production. The parameters of this model are classified as either excitation parameters, related to the source of speech sounds, or vocal tract response parameters related to the individual speech sounds.

The topic of this thesis is related to one of the most important areas in Digital Speech Signal Processing Research, namely, the Digital Coding of Speech Signals.

1.2 DIGITAL SPEECH CODING - ADVANTAGES

Digital coding is used during the transmission of speech signals mainly for the following purposes:-

i) Higher Transmission Efficiency

ii) Simplify Transmitter and Receiver Design

iii) Decrease Errors during Transmission

iv) Provide a Uniform Format for Different Signals

v) Provide Privacy or Secrecy during Transmission

1.2.1 Higher Transmission Efficiency

Digital coding is used on speech signals to achieve higher efficiency during transmission while maintaining the highest possible quality of the transmitted signal. This involves reduction of the channel capacity of the digital transmission link in such a manner that the quality of the signal does not deteriorate below the desired level. By quantising the speech signal to a fixed number of discrete levels, or by employing the differential forms of coding such as the DPAM or DPCM coding schemes, or combinations of both, quantization and differentiation, the transmission rate of the signal can be greatly reduced while maintaining a high signal-to-noise ratio.

1.2.2 Simplify Transmitter and Receiver Design

The transmitter and receiver system used for speech communication can be greatly simplified by employing coding schemes such as Pulse Amplitude Modulation with Quantization, wherein the speech signal is sampled and quantized to assume a fixed number of discrete levels. This results in a lowering of the number of amplitude levels the receiver/transmitter system would have to deal with.

Another coding system used in practice is the PCM or Pulse Code Modulation system wherein the quantized speech signal is coded into strings of binary digits. This system has the great advantage that only the ones and the zeros have to be identified correctly at the other end of the transmission link, unlike the case of an analog system where an infinite number of amplitude levels would have to be transmitted, and received.

1.2.3 Decrease Errors during Transmission

Since the number of amplitude levels, in the case of a quantized speech signal, or the number of states, in the case of a binary coded speech signal is limited to a finite number , the probability of errors being introduced during transmission is greatly reduced. Thus the signal-to noise ratio of the communication system is increased.

1.2.4 Provide a Uniform Format for Different Signals

Digital coding essentially removes the differences in the transmission of speech signals and the transmission of data signals, as far as the communication channel is concerned as once the analog speech signal is sampled and 'discretized', it resembles a data signal as far as the transmission link is concerned. Thus the same transmission link can be used to transmit both digital speech signals, as well as data signals with the only requirement being that at the receiving end, special decoding systems be employed.

1.2.5 To Provide Privacy or Secrecy during Transmission

The digital coding of speech signals is also used for security purposes before transmission to prevent classified information from reaching the hands of unauthorized persons. It also permits extremely sophisticated encryption to be carried out on the speech before and during transmission.

1.3 OBJECTIVES OF RESEARCH

The digital coding of speech signals is employed before transmission to ensure secrecy, in several applications including Defence Communications, and in other voice transmission systems where security is a criteria. In such applications, it may be useful to be able to identify the coding scheme employed in a given coded speech signal obtained perhaps, while monitoring enemy wireless transmissions or by tapping telecommunication lines. In such cases, once the coding scheme used in the coded signal is identified, suitable decoding schemes (in most cases, the inverse operation of the coding scheme) may be used to extract the information content in the given signal.

Another application of code-identification which can be envisaged is in the area of satellite communication. The identification of the coding scheme in a transmitted signal may be useful in a system which automatically allots a channel with the required bandwidth to a user who wishes to use the satellite communication network. The user will just have to transmit a sample length of the coded signal to the system, which, will identify the coding scheme, and after determining the bandwidth required for that code, assign the appropriate transmission channel to that user.

The purpose of this thesis is to carry out an investigation into Speech Code Identification using the techniques of Pattern Classification. The various stages worked on during this investigation are given below:

i) Five important Coding Schemes used in Voice Transmission systems were selected, and implemented in software using different speech sentences. They are:

(a) Pulse Amplitude Modulation (PAM)

(b) First Order Fixed Predictor Differential Pulse Amplitude Modulation

(c) First Order Adaptive Differential Pulse Amplitude Modulation

(d) Second Order Adaptive Differential Pulse Amplitude Modulation

(e) Log-Pulse Amplitude Modulation (Log-PAM) using a U-Law characteristic with U=100

ii) Studies were carried out on the three features selected. The features are:

(a) Short-term Energy Contour

(b) Short-term Zero Crossing Rate Contour

(c) Short-term Normalized Autocorrelation with lag One Contour

The effectiveness of these features in discriminating among the above codes is tested using scatter plots.

iii) Studies were carried out on the derivation of suitable normalization and transformation techniques for improving the effectiveness of the features.

iv) Suitable minimum distance measures were selected for the Classification stage of the Code-Identification System, and modifications were carried out on these measures to improve the identification accuracies.

v) A scheme for identifying the above coding schemes was developed and tested and the results obtained from it are discussed.

1.4 THESIS ORGANIZATION

Chapter 2 explains the principles and the operation of the different stages in a general Pattern Classification system. A block diagram of the proposed Code-Identification System is given, and the reasons for the various components in it are discussed.

Chapter 3 deals with the theory of the different Digital Coding Schemes selected for this study. It also describes the practical considerations taken into account when implementing these codes.

Chapter 4 deals with the Feature Extraction Stage and with the Normalization and Orthogonal Transformation procedures carried out on the features.

Chapter 5 presents the algorithm for the Code-Identification System developed, and the results obtained from it are discussed.

Finally, Chapter 6 presents the Summary and Conclusions of the thesis.

Chapter II

PROPOSED CODE-IDENTIFICATION SYSTEM

2.1 INTRODUCTION

It is the purpose of this thesis to study the feasibility of automatic code-identification in a digitally coded speech signal, using the principles of Pattern Classification. This study would involve derivation of effective features which can discriminate among the coding schemes selected, suitable normalization and transformation schemes for improving the effectiveness of these features, and finally the development of a Classification System for carrying out the identification process using the selected Coding Scheme set.

The techniques of Pattern Classification have been applied in the area of Speech Processing for a variety of applications. These include Speaker Identification, Speaker Verification, Word Recognition and Word Spotting [11]. It is intended to apply some of the concepts and techniques gained from studies of the above Pattern Classification problems to this topic of Code-Identification.

2.2 A GENERAL PATTERN CLASSIFICATION SYSTEM.

The block schematic of a General Pattern Classification System is given in fig. 2.1 [8]. This system consists essentially of the following stages :-

i) Transducer

ii) Feature Extractor

iii) Classifier

2.2.1 Transducer

The transducer in a Pattern Classification System is a device that interfaces the system to the external world.

It senses the input and converts it into a form suitable for machine processing. Typical examples of transducers include cameras for image processing and microphones for speech processing applications. Of course, the term 'transducer' does not strictly refer only to the camera or the microphone, but includes the other stages in the system that may be responsible for transforming the input to a form compatible to the system, like for example, the audio amplifier, tape recorder and A/D converter in the case of a speech processing system, or the video digitiser and disc or magnetic tape storage mechanisms, in the case of an image processing system.

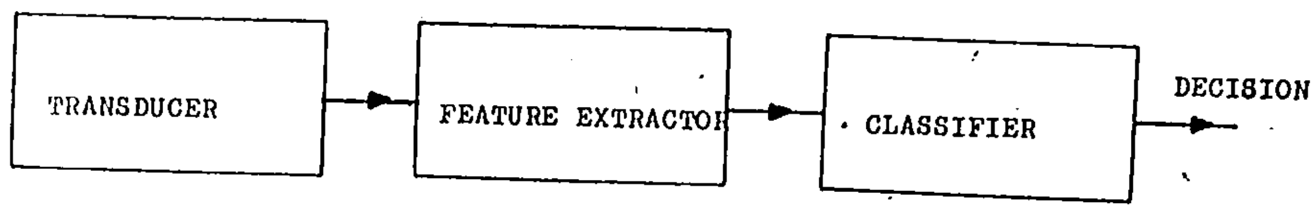


Fig. 2.1 A General Pattern Classification System

2.2.2 Feature Extractor

This is an important stage in the Classification system. The purpose of the feature extractor is to reduce the data derived from the transducer by measuring certain features or parameters that distinguish one class or category of objects to be identified or recognized from another.

The problem of feature extraction is very much dependent on the Classification problem on hand. A good feature extractor for the problem of sorting lumber of different types in a lumber mill would most probably be of little use for identifying finger prints or classifying photomicrographs of blood cells.

2.2.3 Classifier

The classifier stage in a Pattern Classification System uses the information derived from the feature extractorthat is, the features to assign the input data to one of a finite number of categories.

The problem of classification is basically one of partitioning the feature space into regions, one region for each category. Ideally, this partitioning should be carried out in such a way that none of the decisions is ever wrong. However, when this cannot be done, it would be preferred to minimize the probability of error, or if some errors are more costly than others, then, to minimize the average cost of errors. In this case the problem of classification becomes a problem in statistical decision theory.

2.3 PROPOSED CODE-IDENTIFICATION SYSTEM

By using the principles derived from the previous sections on the Pattern Classification Model, the layout of the Code-Identification System is drawn up. The block schematic of the proposed system is as shown in fig.2.2.

A brief description of the various stages is given below. A more detailed description of these stages is given in Chapter 3, Chapter 4 and Chapter 5, where the theory and the practical considerations taken into account when implementing the Digital Coding Schemes, the Features Extraction Stage and the Normalization and Transformation Stages are dealt with. The procedures for forming the test and reference patterns, and the minimum distance metrics used for the classifier stage are also presented.

The operation of the proposed system may broadly be divided into two phases. They are :-

i) The Training Phase

ii) The Testing Phase

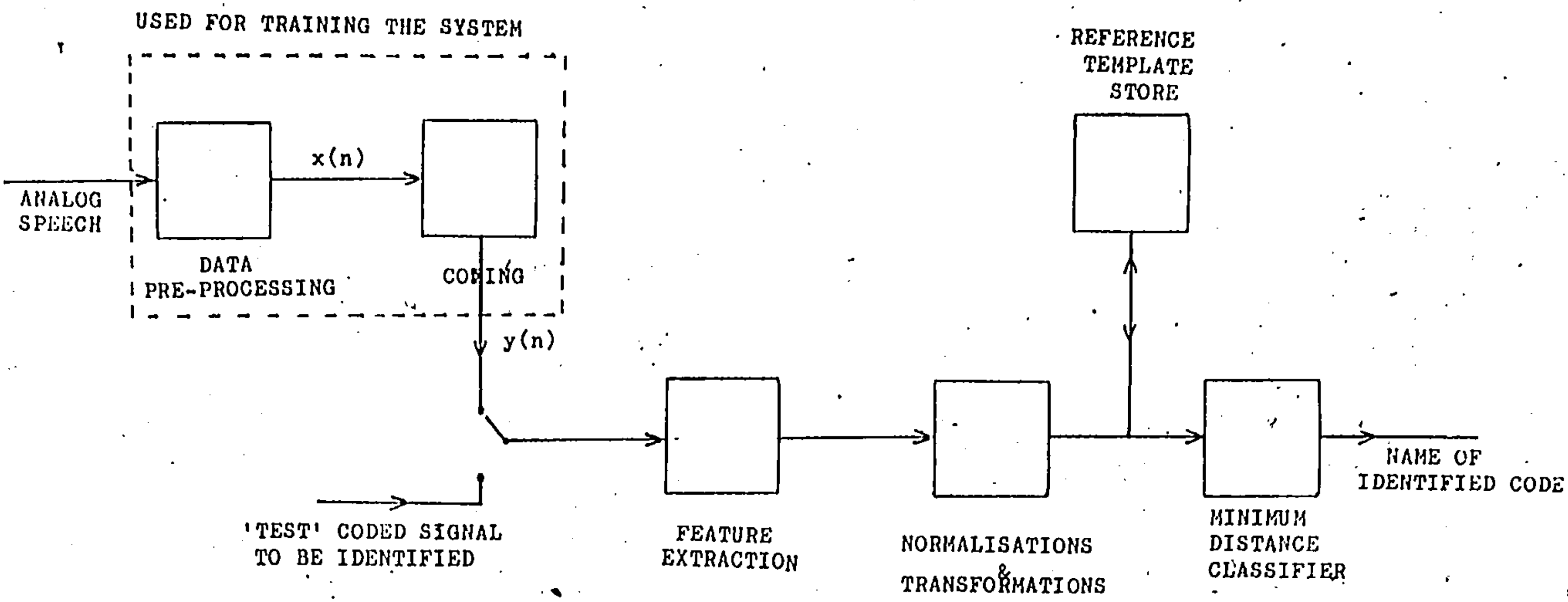


Fig.2.2 Layout of Proposed Code-Identification System

The training phase of the system is used to 'train' the system by generating 'reference' templates or patterns which are stored in the computer memory, and which may later be retrieved for the classification process.

The testing phase of the system is used to generate the test coded signal for testing the accuracy of the identification system.

The training phase of the system consists of the following different operations :-

i) Data Preprocessing

ii) Coding Stage

iii) Feature Extraction Stage

iv) Normalizations and Transformations Stage

v) Reference Template Store

vi) Minimum Distance Classifier Stage

2.3.1 Data Preprocessing

Since the Identification system has to be 'trained' using a number of speech sentences, the function of the data preprocessing stage is to prepare the analog speech sentences for further processing in the computer [7].

The data preprocessing operations include recording different sentences from a variety of speakers using a high quality microphone and tape recorder onto an audio tape. These recorded sentences (about two seconds long) are bandpass filtered with lower and upper cutoff frequencies of 200 Hz and 4.5kHz in order to satisfy the requirements of the Sampling Theorem,

and to remove any extraneous noise in the recordings. The resulting analog signal is then sampled using a 14-Bit A/D convertor and a sampling frequency of 10kHz and stored on the computer disk, for further processing, in separate files 16000 samples long.

2.3.2 Coding Stage

The Coding Stage is used to code the data preprocessed sentences with each of the coding schemes selected for this study. Features can then be selected from each of these coded signals so that reference templates or patterns can be generated for each of these codes and stored for later use in the Classification Stage.

The Coding Schemes selected for this study come under the class of 'Waveform Coders'. These coders essentially approximate the waveshape of the input signal. They are :-

i) Pulse Amplitude Modulation (PAM)

ii) Log-Pulse Amplitude Modulation (Log-PAM) which follows the U-Law Characteristic with U=100

iii) First order Fixed Predictor Differential Pulse Amplitude Modulation

iv) First Order Adaptive Differential Pulse Amplitude Modulation

v) Second Order Adaptive Differential Pulse Amplitude Modulation

A detailed description of these codes is given in Chapter 3.

2.3.3 Feature Extraction

The purpose of the Feature Extraction Stage is to extract meaningful features or parameters from each of the reference coded signals, which can provide information about each code, and which can serve to discriminate one code from another.

Based on the assumption that the speech signal is stationary over short intervals (about 20 ms in duration), short term features are selected, which are computed over each segment, into which the sentences are divided. Since we wish to discriminate one form of coding from another relevant information which could be derived from the coded speech signals about these codes could be in the form of magnitude or energy variations caused by these codes in the time domain, or information about the frequency variations.

The features which have been proposed for this study are:-

i) Short Term Energy Contour

ii) Short Term Zero-Crossing-Rate Contour

iii) Short Term Normalized Autocorrelation With Lag One Contour

2.3.4 Normalizations and Transformations Stage

The purpose of this stage is to carry out normalizations on the raw features, like that of scaling down the magnitudes of the features, or normalizations to make the features sentence-independent by minimising the variability of these features over different sentences.

Transformations for improving the effectiveness of the features by removing the inherent correlations existing among them are also carried out in this stage. The Orthogonal Transformation is used for improving the feature effectiveness. Averaging features obtained from different sentences is another operation which could be carried out in this stage.

2.3.5 Reference Template Store

The reference coded signals from the Coding Stage are passed on to the Feature Extractor which extracts the relevant features, which are then normalized and transformed suitably. These normalized and transformed features can now be stored in the Reference Template Store. This stage stores the reference templates or patterns which uniquely represent each code for later retrieval and comparision with the test-signal-generated pattern for identification in the Classifier Stage.

2.3.6 Minimum Distance Classifier

The Classifier stage in this system is of the 'pattern matching' type. The pattern matching process compares the patterns produced by the incoming test coded signal with the reference patterns stored in the Reference Template Store, and decides on the Classification by detecting the stored pattern which gives the best match. This best match is found by computing a similiarity or dissimiliarity measure between each of the stored patterns and the test-signal generated pattern, and determining which reference pattern gives the highest similiarity value or the lowest dissimiliarity value. The dissimiliarity measure usually takes the form of a minimum distance metric which is computed by taking all the relevant features into account.

Chapter III

DIGITAL SPEECH CODING SCHEMES

3.1 INTRODUCTION

The types of codes considered here for this study come under the class of "Waveform Coders". These codes are essentially straightforward approximations of the time waveforms of the speech signal. They essentially strive for greater coder efficiency by observing the statistics of the given signal set, so that the waveform coder yields minimum encoding error for this signal class i.e. speech. These coders are thus based on the statistical characterization of speech waveforms.

These codes are important because of two reasons.

First, from the point of view of coder complexity, waveform approximating techniques are the most likely candidates for wide-scale applications of Digital Speech Coding [1]. This has been especially true of systems that are required to reproduce speech with a quality sufficient for commercial telephony.

Second, waveform quantization is the most generally applicable approach to coding.

3.2 CHARACTERISTICS OF SPEECH WAVEFORMS

An important characteristic of the speech waveform is that the frequency spectrum falls off rapidly at high frequencies. For most speech processing purposes it is sufficient if the speech signal is bandlimited by lowpass filtering to 4.5kHz, and a sampling frequency of 10 kHz is used to digitise the resulting signal.

Another important characteristic of the speech signal which can be exploited for obtaining coder efficiency is the Amplitude Distribution. The probability distribution function (PDF) of speech amplitudes is characterized ,in general, by a very high probability of zero and near zero amplitudes (related to the unvoiced segments of speech), and by a significant probability of 'very high' amplitudes (for voiced segments) and by a monotonically decreasing function in between these extremes [2]. A coding scheme which takes into account these characteristics of the speech signal can significantly increase the transmission efficiency of the transmission system.

The codes selected for this study can broadly be divided into three classes.

i) Uniform Quantizers- Pulse Amplitude Modulation (PAM)

ii) Nonuniform Quantizers- Log Pulse Amplitude Modulation

iii) Differential Quantization Schemes

The Differential Quantization Schemes can be divided into two kinds :-

i) Fixed Predictor Differential Pulse Amplitude Modulation

ii) Adaptive Predictor Differential Pulse Amplitude Modulation

Under the Fixed Predictor DPAM scheme we select the First Order Fixed Predictor DPAM scheme for this study.

The Adaptive Predictor DPAM schemes, can be divided into two different kinds of Coding Schemes :-

i) First Order Adaptive Predictor DPAM Scheme

ii) Second Order Adaptive Predictor DPAM Scheme

The theory and practical considerations taken when implementing each of these coding schemes is discussed below.

3.2.1 Pulse Amplitude Modulation

In the Pulse Amplitude Modulation of analog speech signals, the analog signal is sampled at regular intervals, corresponding to the interval of the sampling frequency fs, and the amplitude of each sample is rounded off to the nearest one of a finite set of allowable values so that both time and amplitude are in discrete form. This allows the speech signals to be transmitted by means of coded electrical signals in binary form [3], as a Pulse Code Modulated signal.

The generation of a Pulse Amplitude Modulated Signal, (fig.3.2.1(a)), may broadly be divided into two stages :-

i) Sampling

ii) Quantization

3.2.1.1 Sampling

The incoming message wave is sampled with a train of narrow rectangular pulses so as to approximate the instantaneous sampling process. In order to ensure perfect reconstruction of the message at the receiver end, the sampling rate must be greater than twice the highest frequency component W, of the message wave, in accordance with the Sampling Theorem.

In practice, a low pass filter is used at the front end of the sampler in order to exclude frequencies greater than W before sampling. Thus the application of sampling permits the reduction of a continuously varying speech wave to a limited number of discrete values per second.

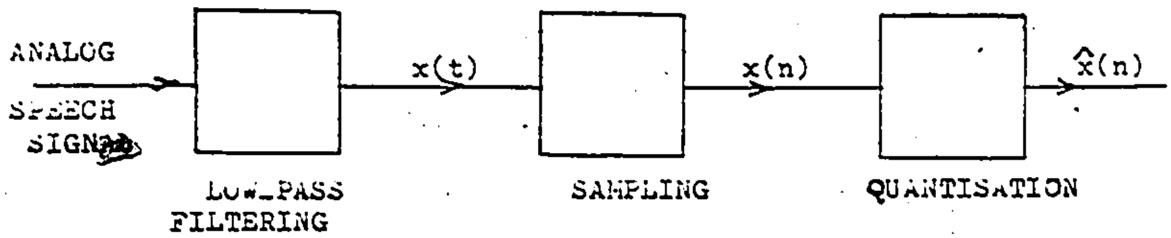


Fig. 3.2.1(a) Pulse Amplitude Modulation System

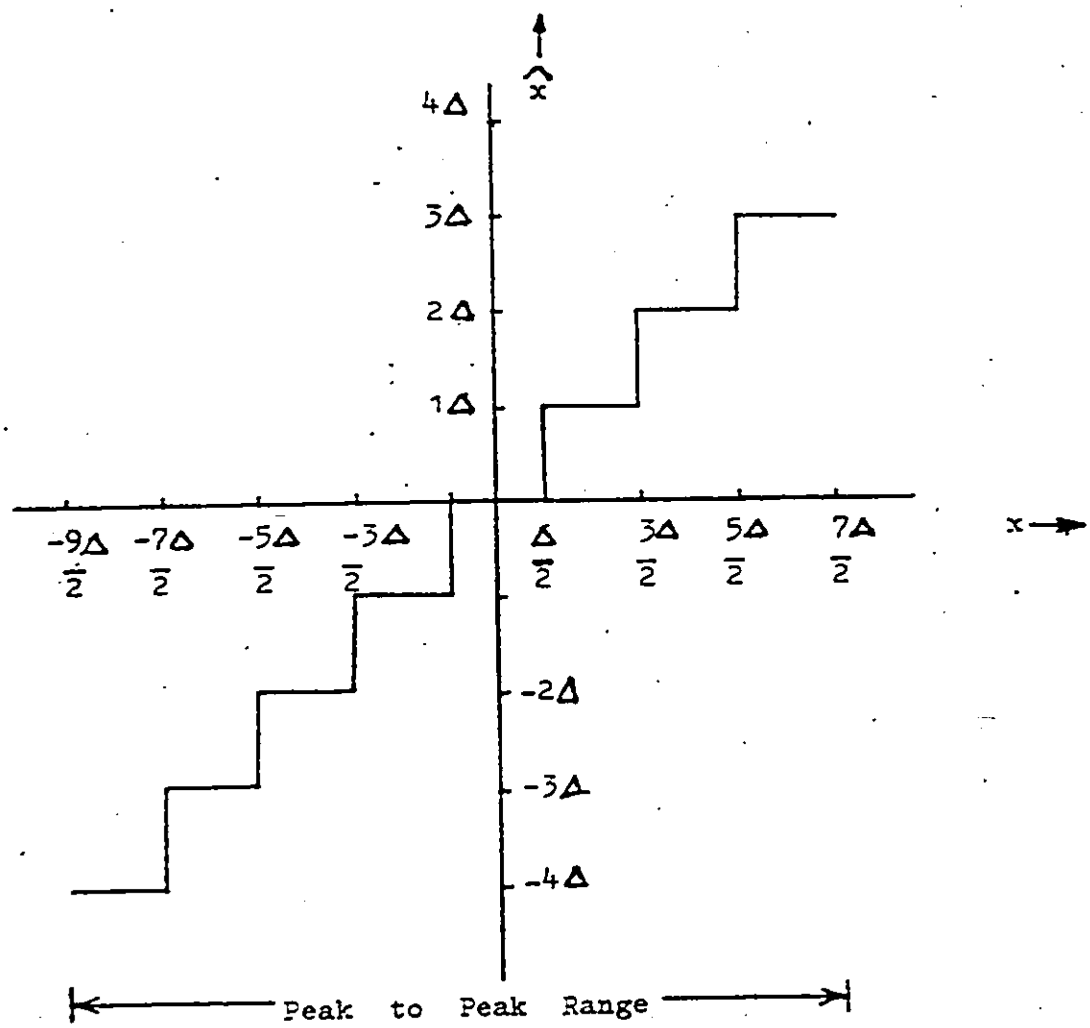


Fig. 3.2.1 (b) Staircase Characteristic of a 3-Bit
Quantiser

3.2.1.2 Quantization

A continuous signal such as speech, has a continuous range of amplitudes and therefore its samples have a continuous amplitude range. This means that within a finite amplitude range of the signal, we find an infinite number of amplitude levels.

Since it is not necessary to transmit the exact amplitudes of the samples, (since the human sense, as the ultimate receiver can only detect finite intensity differences) the orignal continuous signal may be approximated by a signal constructed of discrete amplitudes selected on a minimum error basis from an available set.

This conversion of an analog (continuous) sample of the signal into a digital (discrete) form is called the Quantizing process. Graphically the quantising process may be represented by a staircase characteristic, as shown in fig. 3.2.1(b). The straight line representing the relation between the input and output of a linear continuous system is replaced by a staircase characteristic.

The difference between two adjacent discrete values is called a quantum or the stepsize Δ of the quantizer. Signals applied to the quantizer with the input-output characteristic of fig.3.2.1(b), are sorted into amplitude slices (the treads of the staircase), and all input signals within plus or minus half a quantum step of the midvalue of a slice are replaced in the output by the midvalue in question.

There are two classes of quantizers :-

i) Midrise Quantizers

ii) Midtread Quantizers

Midrise Quantizers have the same number of positive and negative levels and these are symmetrically positioned about the origin.

Midtread Quantizers have one more negative level than positive, though one of the quantization levels is zero. There is no zero level in the midriser case.

The Signal to Noise Ratio of the Uniform Quantizer is given by the equation [1] :

$$SNR(dB) = 6B-7.2$$

where B is the wordlength of each quantized sample.

For the Pulse Amplitude Modulation System implemented in this study, an 8-Bit Uniform Quantizer has been used with a mid-tread type of characteristic.

3.2.2 LOG-PAM (U=100)

The Log-PAM form of coding of speech aims at making the signal to noise ratio of the system independent of the signal value.

When quantizing speech signals, it is preferable to have a variable separation between the quantising levels. The range of amplitudes covered by voice signals, from the peaks of loud talk to the weak passages of weak talk is of the order of 1000 to 1 [3]. By using a non-uniform quantizer with the feature that the step size increases as the separation from the origin of the input-output amplitude characteristic is increased, the large end step of the quantizer can take care of possible excursions of the voice signal into the large amplitude ranges which occur relatively infrequently. The weak passages which 'need more protection' are favoured at the expense of loud passages. In this way a nearly uniform percentage precision is achieved throughout the greater part of the amplitude range of the input signal, with the result that fewer steps are needed than would be the case if a uniform quantizer were used.

The use of a non-uniform quantizer is equivalent to passing the baseband signal through a compressor and then applying the compressed signal to a uniform quantizer, as shown in fig. 3.2.2.(a)

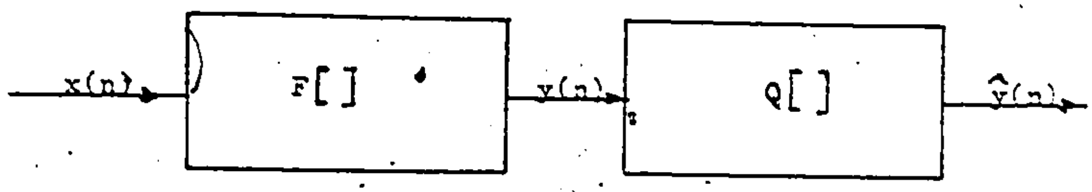


Fig. 3.2.2 (a) Block Schematic of the Log-PAM System (μ=400)

(Compression-Expansion System)

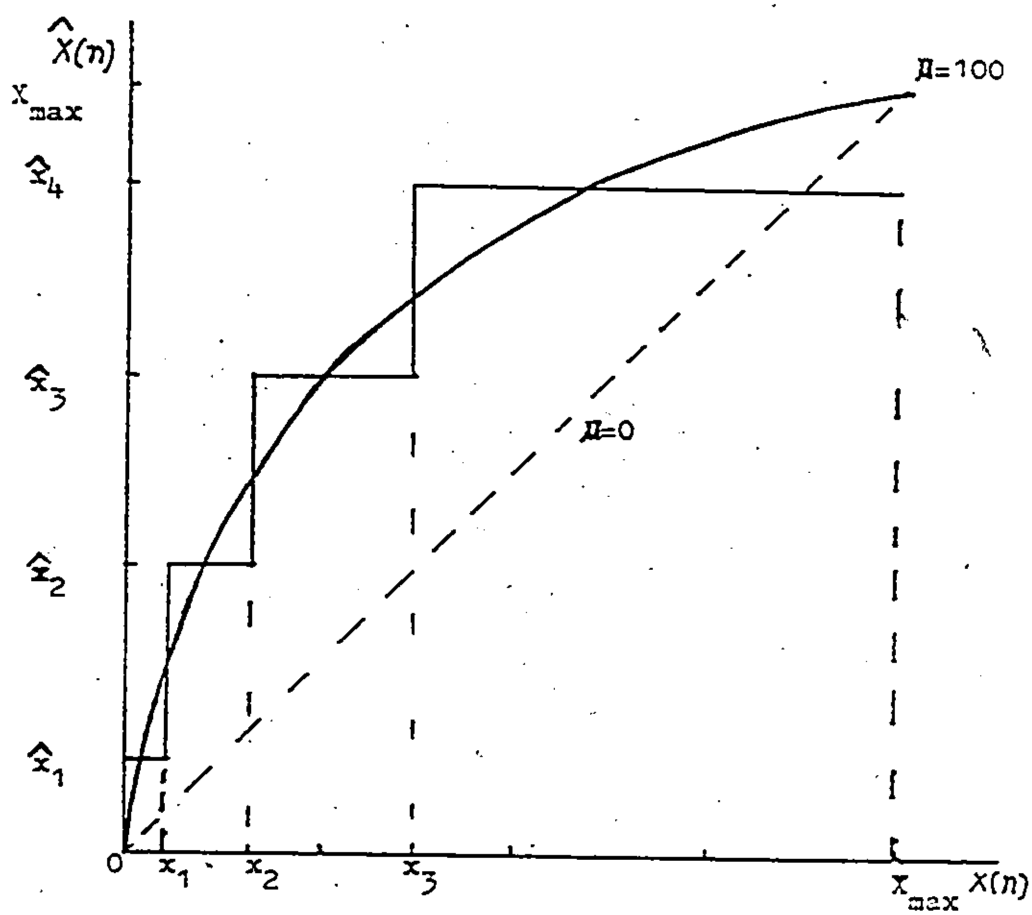


Fig. 3.2.2 (b) Distribution of Quantisation Levels in a μ-Law 3-Bit Quantiser

A particular form of compression law that used in practice is the μ-Law Characteristic [6], defined by,

$$y(n) = F\left[x(n)\right]$$

$$= X_{max} \cdot \frac{\text{Log}\left[1 + \frac{\mu |x(n)|}{X_{max}}\right]}{\text{Log}\left[1 + \mu\right]} \cdot \text{sign}\left[x(n)\right]$$

Typical values of μ used in actual practice are 100 and 255 .

The Signal to Quantizing Ratio for a μ Law quantizer was derived by Smith [6], and shown to be given by the equation :-

$$\text{SNR(db)} = 6B + 4.77 - 20\text{Log}_{10}\left[\ln(1 + \mu)\right]$$

$$- 10\text{Log}_{10}\left[1 + \left[\frac{X_{max}}{\mu \sigma_x}\right] + \sqrt{2}\left[\frac{X_{max}}{\mu \sigma_x}\right]^2\right]$$

where, σ_x is the variance of the speech signal.

For the Log-PAM system implemented in this study, a μ=100 characteristic is chosen with an 8-bit uniform quantizer ~~with~~ mid-tread type of input-output characteristic.

3.2.3 General Differential Quantization Schemes

When a voice signal is sampled at a rate slightly higher than the highest frequency component present in the signal, then the resultant sampled signal is found to exhibit a high correlation between adjacent samples. This correlation is found to be fairly high even between samples that are several sampling intervals apart.

This phenomena shows that in an average sense, the signal does not change rapidly from one sample to the next, with the result that the difference between adjacent samples has a variance that is smaller than the variance of the signal itself. This also shows that the samples contain a significant amount of redundant information.

In order to exploit this sample to sample correlation, the Differential Quantization form of coding is used. A block diagram of this Coding Scheme is shown in fig.3.2.3.

In this system, the input to the quantizer is a signal,

$$d(n)=x(n)-\tilde{x}(n)$$

which is the difference between the unquantized input sample $x(n)$, and an estimate or prediction of the input sample denoted by, $\tilde{x}(n)$.

The predicted value, $\tilde{x}(n)$, is the output of a Predictor system, P, whose input is a quantized version of the input sample x(n). The difference signal may also be called the prediction error signal, since it is the amount by which the predictor fails to exactly predict the input. It can be shown that the quantized speech signal $\hat{x}(n)$, differs from the input only by the quantization error of the difference signal [2].

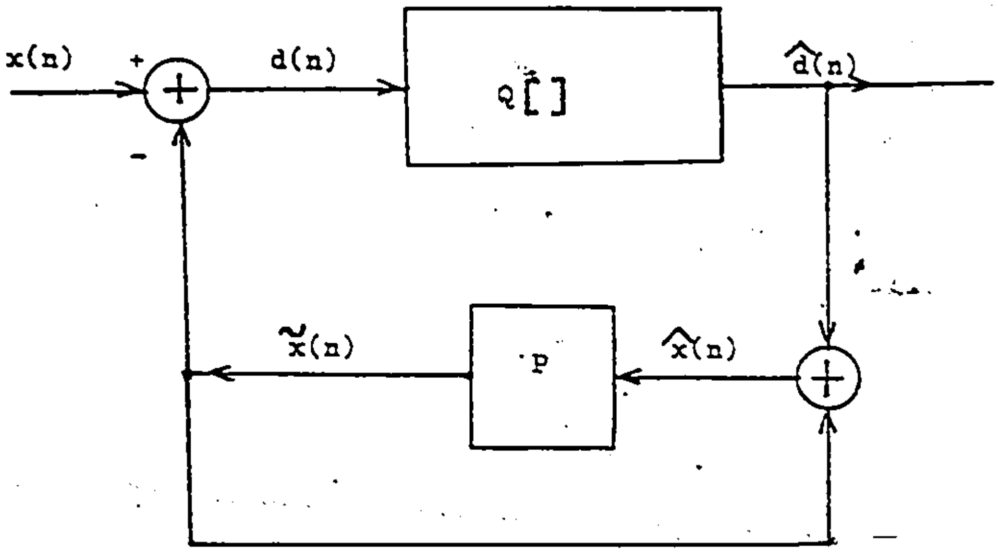


Fig. 3.2.3 General Differential Quantization Scheme

Thus , if the prediction is good, then the variance of d(n) will be smaller than the variance of x(n), so that a quantizer with a given number of levels can be adjusted to give a smaller quantization error than when quanti ing the input directly.

The predictor P can be described by using the linear predictor model of speech. $\tilde{x}(n)$ is thus specified as a linear combination of past quantized values given by:-

$$\tilde{x}(n) = \sum_{k=1}^{P} \alpha_k x(n-k)$$

where P is the order of the finite response filter whose system function is given by the equation :

$$P(z) = \sum_{k=1}^{P} \alpha_k z^{-k}$$

and whose input is the reconstructed quantized signal x(n).

The SNR is dependent on the particular quantizer that is used, and given the knowledge of the properties of d(n), the SNR can be maximized by minimizing the variance of the difference, that is, by minimizing the variance of the prediction error.

This is the philosophy behind the Adaptive Differential Quantization Schemes.

Thus it can be concluded that Differential Quantization can yield improvements over Direct Quantization.

Second, the amount of improvement is dependent upon the amount of correlation in the input signal.

Third, a fixed predictor cannot be optimum for all speech materials, thus making it necessary to have adaptive schemes, whereby adaptive predictive coefficients are derived over short segments of the speech signals, and used for the coding.

The three forms of Differential Quantization Schemes considered in this study are :-

i) First Order Fixed Predictor DPAM

ii) First Order Adaptive Predictor DPAM

iii) Second Order Adaptive Predictor DPAM

3.2.3.1 First Order Fixed Predictor DPAM

In this form of coding, the order of the predictor filter is one. The equation of the Prediction filter is given by:

$$\tilde{x}(n) = a_1 x(n-1)$$

where a_1 is a constant.

For this study, a_1 is chosen to be equal to 1 and the quantizer used is an 8-Bit Uniform Quantizer with a mid-tread characteristic.

3.2.3.2 Adaptive Prediction DPAM

Fixed Prediction Differential Schemes give under the best circumstances, 10-12 db improvement in the SNR over the Non-Predictive Quantization schemes.

Further more, the improvement in the SNR is a function of the speaker and of the speech material. In order to effectively cope with the non-stationarity of the speech communication process, the process of adapting the predictor to match the temporal variations of the speech signal is also considered.

Here the speech sample is divided into short segments (about 20 mS in duration, over which the speech signal can be considered to be stationary) and the predictor coefficients α_k are computed over these segments by minimizing the average squared prediction error, and are used in computing the predicted signal $\tilde{x}(n)$, given by :-

$$\tilde{x}(n) = \sum_{k=1}^{P} \alpha_k \hat{x}(n-k)$$

There are two forms of Adaptive Predictor Schemes. They are the Feedforward Adaption Method and the Feedback Adaption Method.

In the feedforward scheme, the predictor coefficients are based upon measurements on the input signal, whereas in the case of the feedback system, the prediction coefficients are derived from the quantised signal $\hat{x}(n)$.

It can be shown that the predictor coefficients satisfy the equations :-

$$R(j) = \sum_{k=1}^{P} \alpha_k R(j-k) \qquad j=1,2,----,P$$

where $$R(j) = \sum_{m=-\infty}^{\infty} x(n)w(n-m)x(j+m)w(n-m-j)$$

$$0 \leqslant j \leqslant P$$

and $w(n-m)$ is a window that is positioned at sample n of the input sequence.

3.2.3.3 First Order Adaptive Predictor DPAM

Thus for the First Order Adaptive Predictor DPAM system, we substitute P=1 in the above equations. We thus get the equation for the First Order Adapter Predictor,

$$\alpha_1 = \frac{R(1)}{R(0)}$$

where R (1) is the autocorrelation with lag one
and R (0) is the short term energy.

The derivation of this equation is given in Appendix- A.

For this study, the length of each segment over which the autocorrelation coefficients, and hence the prediction coefficient is calculated is chosen to be 200 samples long. The quantiser used in this case is an 8-Bit Uniform Quantiser with a mid-tread characteristic.

3.2.3.4 Second Order Adaptive DPAM

For the Second Order Adaptive DPAM scheme, we substitute P=2 in the equation for the prediction filter thus obtaining the equation for the prediction signal as

$$\tilde{x}(n) = \alpha_1 \hat{x}(n-1) + \alpha_2 \hat{x}(n-2)$$

In the implementation of this coding scheme, the autocorrelation coefficients, R(0), R(1) and R(2) are computed over segments of 200 samples each. The quantiser used is an 8-Bit Uniform Quantiser with a midtread characteristic.

$$\tilde{x}(n) = \alpha_1 \hat{x}(n-1) + \alpha_2 \hat{x}(n-2)$$

The difference signal is thus given by:

$$d(n) = x(n) - \alpha_1 \hat{x}(n-1) - \alpha_2 \hat{x}(n-2)$$

α_1 and α_2 can be shown to be given by the equations:

$$\alpha_1 = \frac{R(0)R(1) - R(1)R(2)}{R^2(0) - R^2(1)}$$

$$\alpha_2 = \frac{R^2(1) - R(2)R(0)}{R^2(1) - R^2(0)}$$

where $R(0)$ is the short-term energy over the segment
$R(1)$ is the autocorrelation with lag one over the segment.
$R(2)$ is the autocorrelation with lag two over the segment.

The derivation of these equations is given in Appendix- A.

Chapter IV

FEATURE EXTRACTION & NORMALIZATIONS/TRANSFORMATIONS

4.1 INTRODUCTION

This chapter deals with the feature extraction and normalizations/ transformations stages of the Code-Identification System being studied.

The three features selected for this study are described in this chapter, and the various normalization and transformation procedures carried out on these features with the purpose of improving their code-discrimination properties are explained.

The results of the feature effectiveness tests carried out on the normalized and transformed features using scatter plots are also given.

4.2 FEATURE EXTRACTION

Feature Extraction is one of the most important stages in the identification process. The purpose of this stage is to derive meaningful features or parameters from the coded speech signals, which can effectively discriminate one code from another, when plotted on a multidimensional feature space (the number of dimensions depending on the number of features being used).

These features should not only discriminate between the codes, but should also be independent of the sentences and the speakers uttering them.

The coding of speech signals affects the amplitudes and the frequency components of the basic speech signal. Therefore, in order to characterize a particular code, the features should provide information about these variations of amplitudes and frequency.

The features selected for this study based on these requirements are :-

i) Short-Term Energy Contour
ii) Short-Term Zero Crossing Rate Contour
iii) Short-Term Normalized Autocorrelation With Lag One

These features are basically short term features which are computed over short segments of the coded speech signal. These short segments over which these features are computed are chosen to be 20 ms in duration, or 200 samples long for a sampling frequency of 10kHz, so that the assumption that the coded speech signal is stationary over these intervals is valid.

4.2.1 Short-Term Energy Contour

The Short-Term Energy Contour is obtained by extracting the short-term energy over each segment of the coded sentence (the length of each segment depending on the frame-size or segment length being chosen) and plotting the energy points as a continuous waveform. The short-term energy provides us with a convenient representation of the amplitude variations of the coded signal.

The short-term energy is defined by the equation:

$$E = \sum_{m=-\infty}^{\infty} \left[x(m)w(n-m)\right]^2$$

This can also be written as:

$$E = \sum_{m=-\infty}^{\infty} x^2(m)h(n-m)$$

where $h(n) = w^2(n)$

$h(n)$ represents the window over which the short-term energy is computed.

For our short-term energy computation, a rectangular window $h(n)$ is chosen, of the form:

$$h(n) = \begin{cases} 1, & 0 \leqslant n \leqslant N-1 \\ 0, & \text{otherwise} \end{cases}$$

where N represents the frame size of the speech segments.

4.2.2 Short-Term Zero Crossing Rate Contour

The zero crossing rate is a simple measure of the frequency content of a signal. A zero crossing is said to occur if successive samples have different algebraic signs.

4.2.2 Short-Term Zero Crossing Rate Contour

The zero crossing rate is a simple measure of the frequency content of a signal. A zero crossing is said to occur if successive samples have different algebraic signs. The short-term zero crossing rate is defined by the equation:

$$Z = \sum_{m=-\infty}^{\infty} \left| \operatorname{sgn}\left[x(m)\right] - \operatorname{sgn}\left[x(m-1)\right] \right| w(n-m)$$

where $\operatorname{sgn}[x(n)] = 1 \quad x(n) \geq 0$
$\qquad\qquad = -1 \quad x(n) < 0$

and $w(n) = 1/2N \qquad 0 \leq n \leq N-1$
$\qquad\quad = 0 \qquad$ otherwise

This feature could provide a good representation of the coding scheme employed as it gives an indication of the frequency variations of the signal over which it is computed. This feature crudely reflects the spectral properties of the signal.

4.2.3 Short-Term Normalized Autocorrelation with Lag One

The normalized autocorrelation with lag one is found by computing the short term autocorrelation R(k) over each segment of the coded speech signal. The short term autocorrelation is defined by the equation :

The normalized autocorrelation with lag one, C(1), is obtained by extracting the autocorrelation coefficient R(1)

$$R(k) = 1/(N-k) \cdot \sum_{m=0}^{N+k-1} x(m)x(m+k)$$

where k is the lag and is equal to one for deriving this feature.

for each segment, and normalising (dividing) each coefficient by the energy R(0).

$$C(1) = R(1)/R(0)$$

This feature also gives a good indication of the rate at which the signal varies. The values C(1) can take are constrained to the range [-1,+1] , negative values indicating that the rate of variation is high and positive values indicating that the rate of variation of the signal is low.

4.3 TESTING OF FEATURES BY SCATTER PLOTS

The effectiveness of the features extracted by the Feature Extraction stage in discriminating one code from another is tested by the use of scatter plots.

These plots are obtained by plotting one feature against another on a two-dimensional feature space. If a natural grouping of points belonging to each code is produced, each group being well separated and distinct from the others, then the usefulness of the feature in discriminating one code from another is confirmed. The plotting is carried out for each combination of two features, for each sentence.

The following scatter plots are obtained using the features described above:

i) Energy Contour vs. Zero Crossing Rate

ii) Energy Contour vs. Normalized Autocorrelation with Lag One

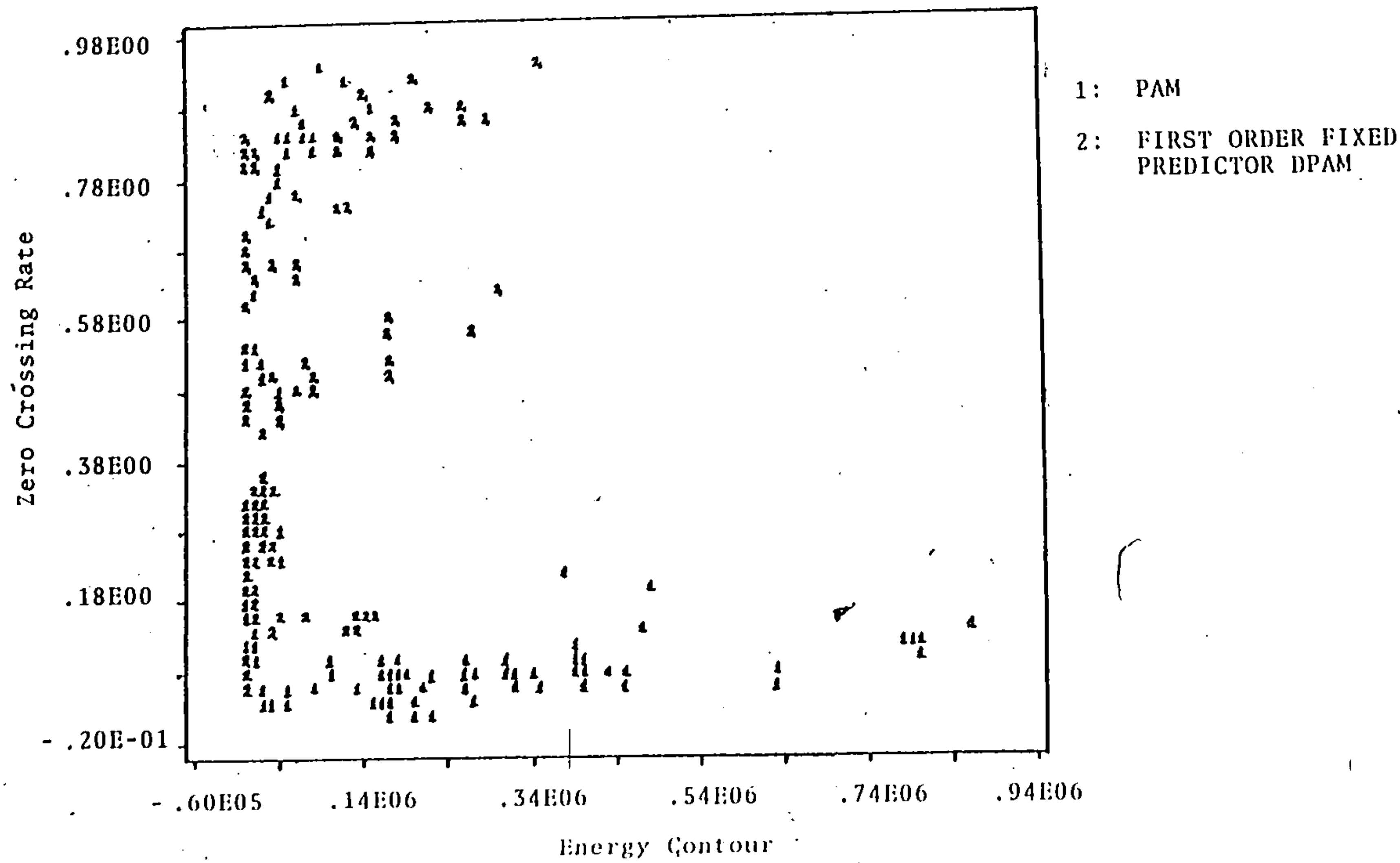
iii) Zero Crossing Rate vs. Normalized Autocorrelation with Lag One

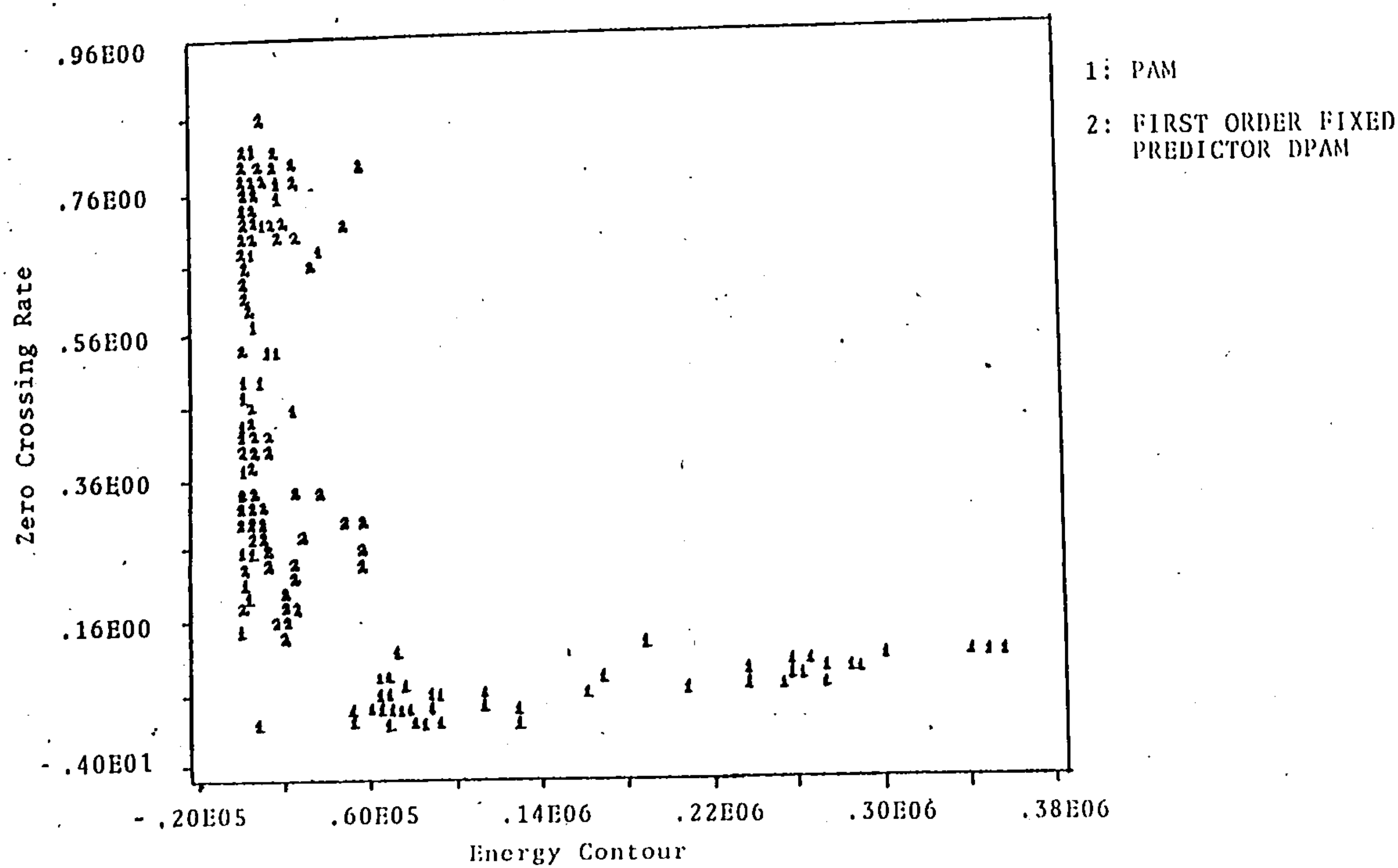
An example of the scatter plot obtained by plotting the 'basic' features, Energy Contour versus Zero Crossing Rate for the two codes, Pulse Amplitude Modulation , and First Order Fixed Predictor DPAM, on the same speech sentence, is shown in fig. 4.1.

From this scatter plot it can be seen that the two features are fairly effective in discriminating between the two codes, through the formation of naturally forming clusters or groups of feature points.

However, a slight compression of points along the Energy Axis of the plot can be noticed.

Fig. 4.2 shows the scatter plot obtained by plotting the Energy Contour against the Zero Crossing Rate for the same two codes, Pulse Code Modulation and First Order Fixed Predictor DPAM using a different speech sentence, 'SE2'. From this plot it can be seen that the clustering of feature points for the two codes have similiar shapes and distributions as those in the previous plot (fig. 4.1). However, on observing the numerical values of the features on the two axes, it can be seen that the clusters formed by the same codes are at different locations in the feature space formed by the Energy Contour and Zero Crossing Rate, for different sentences. This shows that though the features are code-discriminating over a coded speech sentence, they are dependent on the sentence being used.


Zero Crossing Rate
.98E00
.78E00
.58E00
.38E00
.18E00
-.20E-01
-.60E05
.14E06
.34E06
.54E06
.74E06
.94E06
Energy Contour
1: PAM
2: FIRST ORDER FIXED PREDICTOR DPAM



Zero Crossing Rate
.96E00
.76E00
.56E00
.36E00
.16E00
-.40E01
-.20E05
.60E05
.14E06
.22E06
.30E06
.38E06
Energy Contour
1: PAM
2: FIRST ORDER FIXED PREDICTOR DPAM

Another point which can be observed from the scatter plots using different pairs of features is that not all features have the same level of effectiveness in discriminating between the codes.

The order of effectiveness of the feature-pairs is found to be as given below:

(1) Energy Contour vs. Zero Crossing Rate (Maximum Code-Discrimination)

(2) Energy Contour vs. Normalized Autocorrelation With Lag One (Next best in Code-Discrimination)

(3) Zero Crossing Rate vs. Normalized Autocorrelation with Lag One (Least Code-Discrimination)

Thus it was found necessary to devise suitable Normalization and Transformation Schemes to tackle the problems observed from the scatter plots, namely,

(1) Compression of points along the Energy Contour Axis

(2) Improve the Code-Discrimination Properties of the three features

(3) Transforming the three features so that they are rendered Sentence-Independent

The Normalization and Transformation Schemes carried out on the features are:

i) Scaling the Energy Points

ii) Orthogonal Transformation

iii) Normalization of Orthogonal Parameters to Unit Variance

iv) Averaging of Features over Many Sentences

4.2.1 Scaling of Energy Points

Scaling of Energy points was found necessary due to the very high numerical values (of the order of 2×10^5 J) in comparison with the numerical values of the other features (of the order of 10^{-1}).

The scaling scheme involved the normalizing of the energy points by dividing each energy point by a constant value C. The constant value chosen for this purpose was 10^5 . This form of normalization was found to remove the compressive tendency of the points along the Energy Axis, and did not affect the code-discrimination properties of the features.

This scheme can be described by the equation:

$$EN(m)=E(m)/C$$

$$m=1,2,\text{-------}NF$$

where,

NF is the number of segments in a sentence (= 80)

E(m) is the short term energy of the m th segment

EN(m) is the normalized short term energy of the m th segment

and, C is a constant (= 10^5)

4.3.2 Orthogonal Transformation

The next operation carried out on the normalized feature set is the Orthogonal Transformation.

The orthogonalization of features essentially removes the correlations existing between the features, and generates mutually uncorrelated orthogonal parameters. These orthogonal parameters are thus more code-sensitve than the original feature set, and are thus more effective in discriminating between the codes.

The different steps involved in the calculation of the orthogonal parameters are as given below [9] :-

(1) Let x_{ij}: i=1,2,----M; j=1,2,----NF be the feature set, where x_{ij} is the i th parameter of the j th frame; M is the number of features in the set (=3)

and NF is the total number of analysis frames in the coded sentence.

(2) Compute the Covariance Matrix of the feature set, where $[C] : c_{lm} : l=1,2,----, M; m=1,2,----, M$ is given by:

$$c_{lm} = 1/(NF-1)\sum_{j=1}^{NF}(x_{lj} - \bar{x}_l)(x_{mj} - \bar{x}_m)$$

and,

$$x_l = (1/NF)\sum_{j=1}^{NF} x_{lj}$$ is the average value of the l th parameter.

(3) Compute the eigen values $\lambda_l : l=1,2,----,M$ and the eigen vectors T_l of the matrix $[C]$ by solving the equation, $C- \quad I = 0$ for $_l$' s and by solving the equation $CT_l=\lambda_l T_l$ for T_l .

(4) Normalize T_l to unit length.

(5) Evaluate the Orthogonal Parameters ($\phi_{ij} : i=1,2,----,M;$ $j=1,2,----,NF$) as follows:

$$\phi_{ij}=\sum_{l=1}^{M} t_{il}x_{lj}$$

where, ϕ_{ij} is the i th Orthogonal Parameter in the j th frame;

t_{il} is the I th element of the i th eigen vector T_i .

4.3.3 Normalisation to Unit Variance

The next step in the normalisation process after Orthogonal Transformation, is the normalisation of each orthogonal parameter to unit variance.

This operation is carried out by computing the standard deviation of each feature over a sentence, and dividing each feature point by that features's standard deviation [8].

The equation for this normalisation scheme is given below :-

$$\emptyset N_{ij} = \emptyset_{ij} / \left[(1/NF) \sum_{k=1}^{NF} (\emptyset_{ik} - \bar{\emptyset}_{i})^{2} \right] \quad i = 1,2,----,M$$

$$j = 1,2,----,NF$$

where ,

M is the number of orthogonal parameters (=3)

NF is the number of frames in one sentence (=80)

$\emptyset N_{ij}$ is the i th normalised orthogonal parameter in the j th frame

and $\emptyset_{ij}$ is the i th orthogonal parameter in the j th frame

and $\bar{\emptyset}_{i}$ is the mean of the i th orthogonal parameter derived over NF frames.

4.3.4 **Averaging**

From the scatter-plots in fig.4.1 and fig.4.2, it was observed that the clusters formed for a particular code from different sentences, occupied different locations in the feature space. However, the shape of the clusters and the distribution of the points for both clusters were similiar.

Two different methods of averaging the normalised orthogonal parameters were conceived and carried out, and the scatter plots showing the improvements in the code-discrimination properties of the averaged parameters were derived.

The two averaging schemes carried out are :-

i) Averaging the parameters over many sentences

ii) Averaging the parameters over each sentence, for many sentences

4.3.4.1 Averaging the Parameters over Many Sentences

A method of averaging the clusters obtained from different sentences in order to generate a 'reference' cluster for a particular code for storage in the Reference Template Store, was conceived.

This method can be described by the equation :-

$$\varnothing NAV1_{ij} = 1/NFL \sum_{k=1}^{NFL} \varnothing N_{ijk} \qquad \begin{array}{l} i = 1,2,\text{---}, M \\ j = 1,2,\text{----}, NF \end{array}$$

where,

N is the number of parameters (= 3)

NF is the number of frames in one sentence (= 80)

NPL is the number of sentences chosen for the averaging.

$ØNAV_{ij}$ is the i th average orthogonal parameter of the j th segment.

$ØN_{ijk}$ is the i th orthogonal parameter derived over the j th segment of the k th sentence.

Figures 4.3.4.1(a) and 4.3.4.1(b) show the scatter plots obtained by plotting the first and the second averaged orthogonal parameters on the two axes for different pairs of codes.

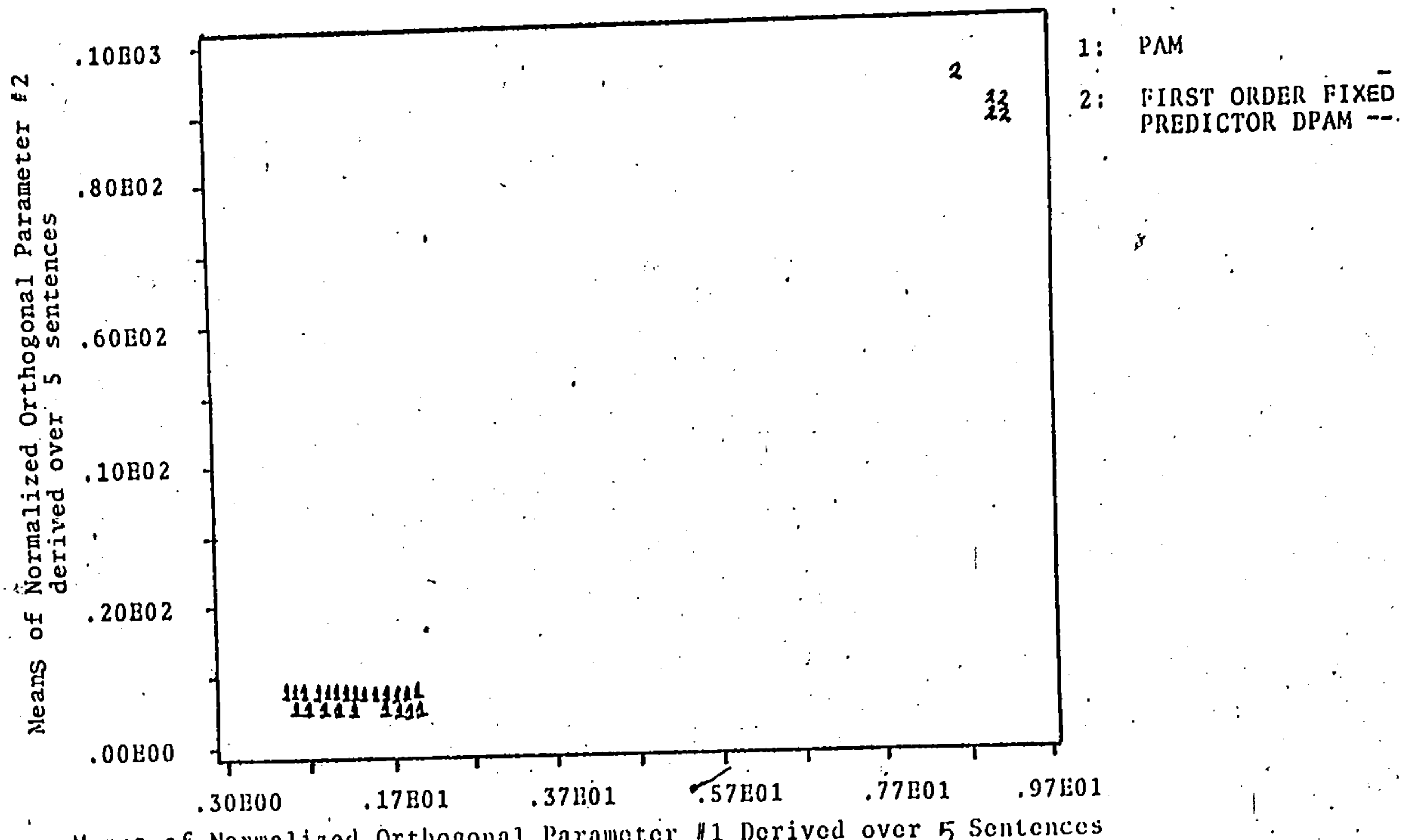


Fig.4.3.4.1(a) Scatter Plot of the Means of the Normalised Orthogonal Parameter #1 vs. the Means of the Normalised Orthogonal #2 for the PAM and First Order Fixed Predictor DPAM codes (Averaging over 5 Files).

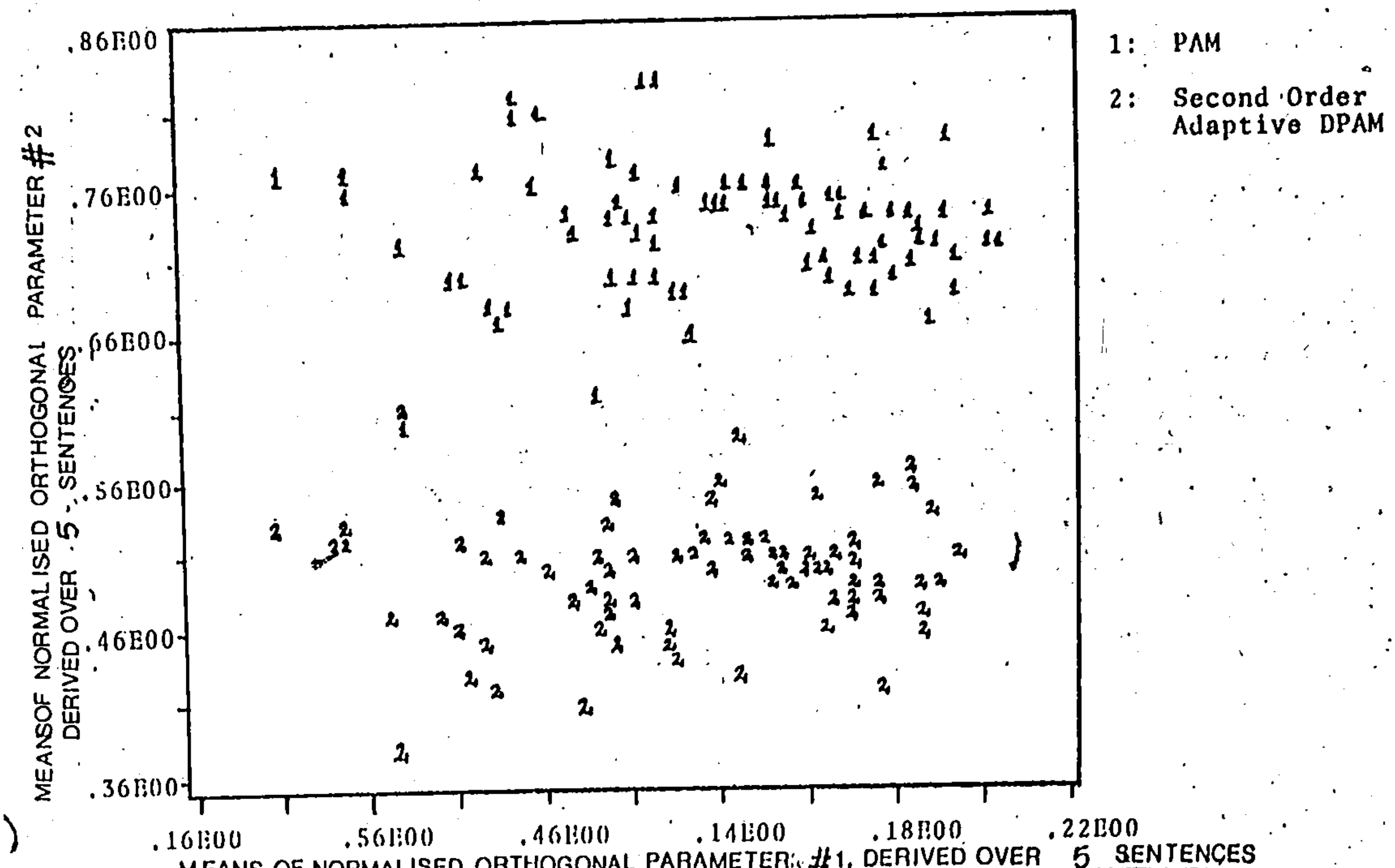


Fig.4.3.4.1(b) Scatter Plot of the Means of the Normalised Orthogonal Parameter #1 vs. the Means of the Normalised Orthogonal Parameter #2 for the PAM and Second Order Adaptive Predictor DPAM codes(Averaging over 5 Files).

Only two of the plots are included here for illustrative purposes.

From these plots, it can be seen that the averaged normalized orthogonal parameters form very clear and distinct clusters for each of the codes. The separation between the clusters varys, depending on the pair of codes being considered in the scatter plot.

In the case of the scatter plot showing the points obtained from the Pulse Amplitude Modulation and the Second Order Adaptive Predictor DPAM codes, it acn be seen that the clusters formed are very close to each other, and, in fact there are some areas of overlap between the two clusters. This indicates that clusters formed by the two codes are located very closely in the feature space formed by the First and Second Averaged Normalized Orthogonal Parameters, thus increasing the probability of misidentification of one of the codes for the other, if the same feature space is used for identification purposes.

4.2.1.1 Averaging the Parameters over each Sentence

Another process of averaging the parameters was carried out wherein the average of each normalized orthogonal parameter over one coded sentence was computed. Similiar averages of each normalized orthogonal parameter were computed for five different coded sentences, and were plotted one against the other on scatter plots.

This averaging process can be described by the equation;

$$ØNAV2_{ik} = \left[\sum_{j=1}^{NF} ØN_{ijk}\right] \Big/ NFL$$

$$i=1,2,---,M$$

$$k=1,2,---,NFL$$

where,

M is the number of features (=3)

NFL is the number of coded speech sentences

NF is the number of frames in one sentence

$ØN_{ijk}$ is the i th normalised orthogonal parameter of the j th frame of the k th coded sentence

$ØNAV2_{ik}$ is the average of the i th normalised orthogonal parameter of the k th coded sentence.

Figures 4.3.4.2(a) and 4.3.4.2(b) show the scatter plots obtained by plotting the means of the normalised orthogonal parameters for the five codes. Each point in the plot represents the mean of the normalised orthogonal parameter derived over one sentence for one code.

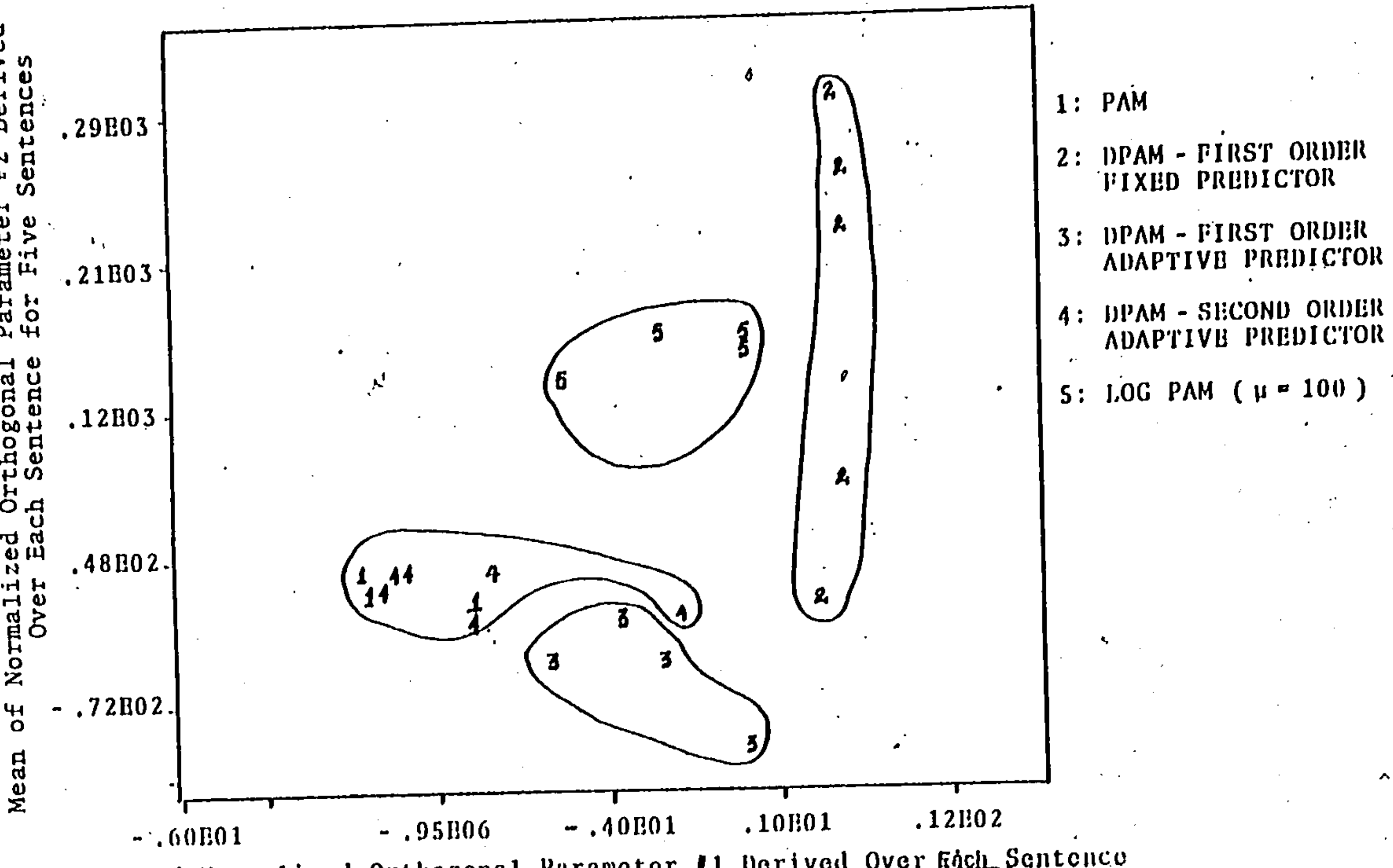


Fig-4.3.4.2(θ) Scatter Plot of the Means of the Normalised Orthogonal Parameter #1 vs. the Means of the Normalised Orthogonal Parameter #2 for all the five codes. (averaging over 1 sentence for five sentences).

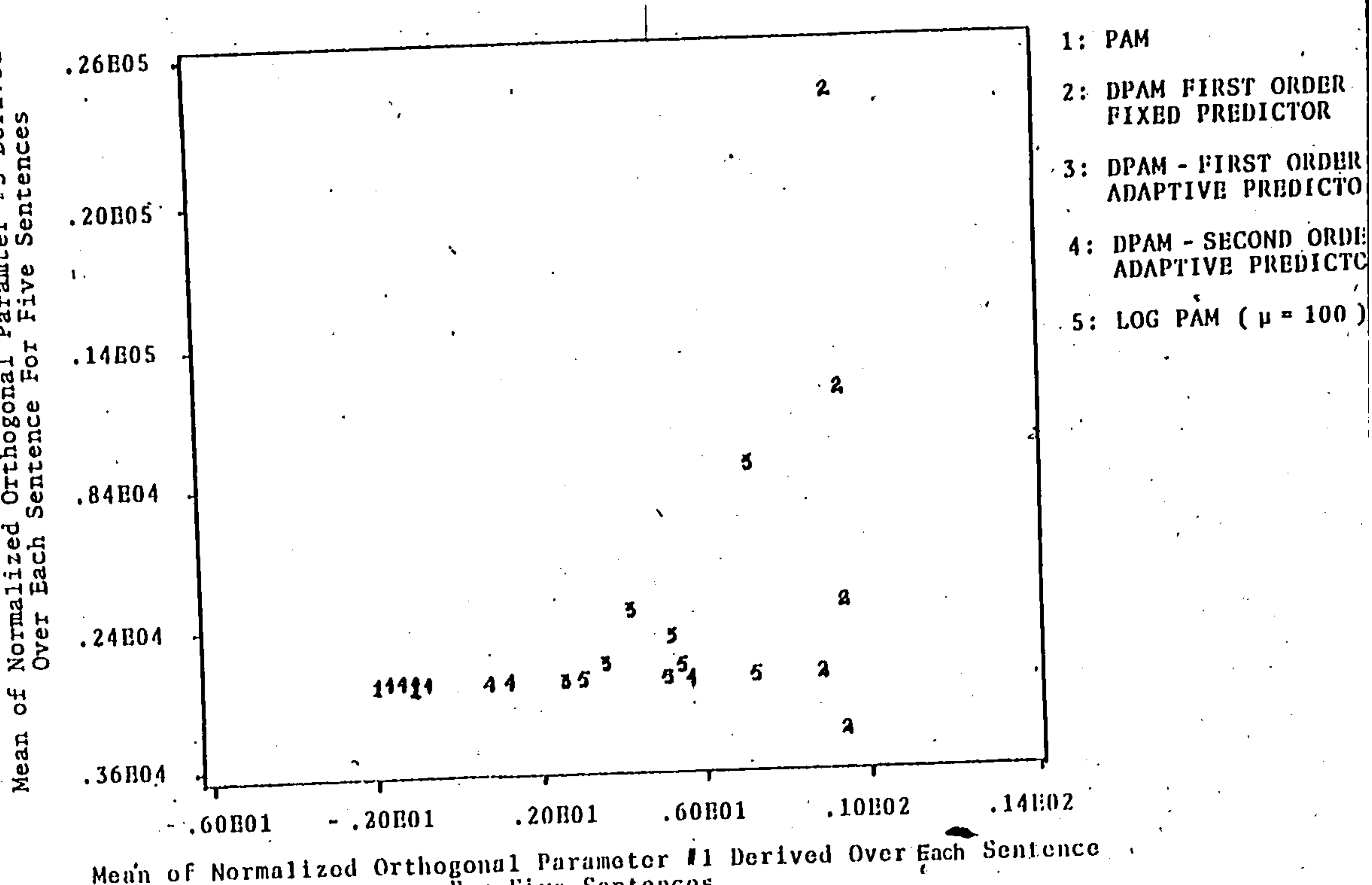


Fig. 4.3.4.2(b) Scatter Plot of the Means of the Normalised Orthogonal Parameter #1 vs. the Means of the Normalised Orthogonal Parameter #3 for all the five codes. (averaging over 1 sentence for 5 sentences).

Figure 4.3.4.2(a) shows the scatter plots obtained by plotting the means of the first and the second normalized orthogonal parameters for the five codes.

From this scatter plot it can be seen that three of the codes, namely, First Order Fixed Predictor DPAM, First Order Adaptive Predictor DPAM and Log-PAM form separate distinct clusters. However, the points obtained from Pulse Amplitude Modulation and Second Order Adaptive Predictor DPAM are intermingled showing that the two codes are more susceptible to misidentification than the other codes.

On observing the figure 4.3.4.2(b), it can be seen that the points obtained by plotting the means of the first and the third normalized orthogonal parameters for the five codes are intermingled, showing that the third normalized orthogonal parameter does not contribute much towards discriminating between the codes.

Chapter V

CLASSIFICATION ALGORITHMS & RESULTS

5.1 GENERAL

This chapter deals with the final stage - the Classification stage -in the proposed Code-Identification System. Classification is basically a process of comparing the test-signal pattern with the stored reference pattern of each of the codes selected for this study. The reference pattern which gives the closest match to the test signal generated pattern is taken as the identified code. The pattern matching procedure is carried out in the classifier by computing a dissimiliarity measure using the normalized and transformed features.

In this chapter, an algorithm which derives the reference patterns and test patterns and which uses the orthogonal distance metric for the classification is presented.

For this algorithm, the data set consists of two parts :-

(1) The Training Set

(2) The Testing Set

The training set consists of five files consisting of five different sentences, and the test set consists of another set of five files consisting of five other sentences.

5.2 ALGORITHM #1

Based on the results of the scatter plots described in the previous chapter (figures 4.3.4.1(a) and 4.3.4.1(b)) obtained from the first averaging scheme, 'Averaging the Parameters over many Sentences', (Section 4.3.4.1), the following algorithm was developed. The flowchart of this algorithm is given in fig. 5.3. This system essentially consists of the following phases of operation :-

i) The Training Phase

ii) The Testing Phase

iii) The Classification Phase

5.2.1 The Training Phase

The training phase of Algorithm #1 consists of the Coding Stage, the Feature Extraction and Scaling Stage, and finally the Averaging Stage.

The Coding Stage codes each of the five reference Speech files separately with the five coding schemes, and the Feature Extraction stage extracts the short-term features of Energy Contour, Zero Crossing Rate and Normalized Autocorrelation with Lag One from each of these coded speech sentences.

The Energy points are scaled by dividing each energy point by a constant equal to 10^5.

The Averaging Stage now computes the mean value of each of these scaled features by summing up their values obtained from all the five sentences and dividing these sums by five. We thus obtain a reference feature vector for each code consisting of three values representing the mean values of the three features over the five reference coded speech sentences.

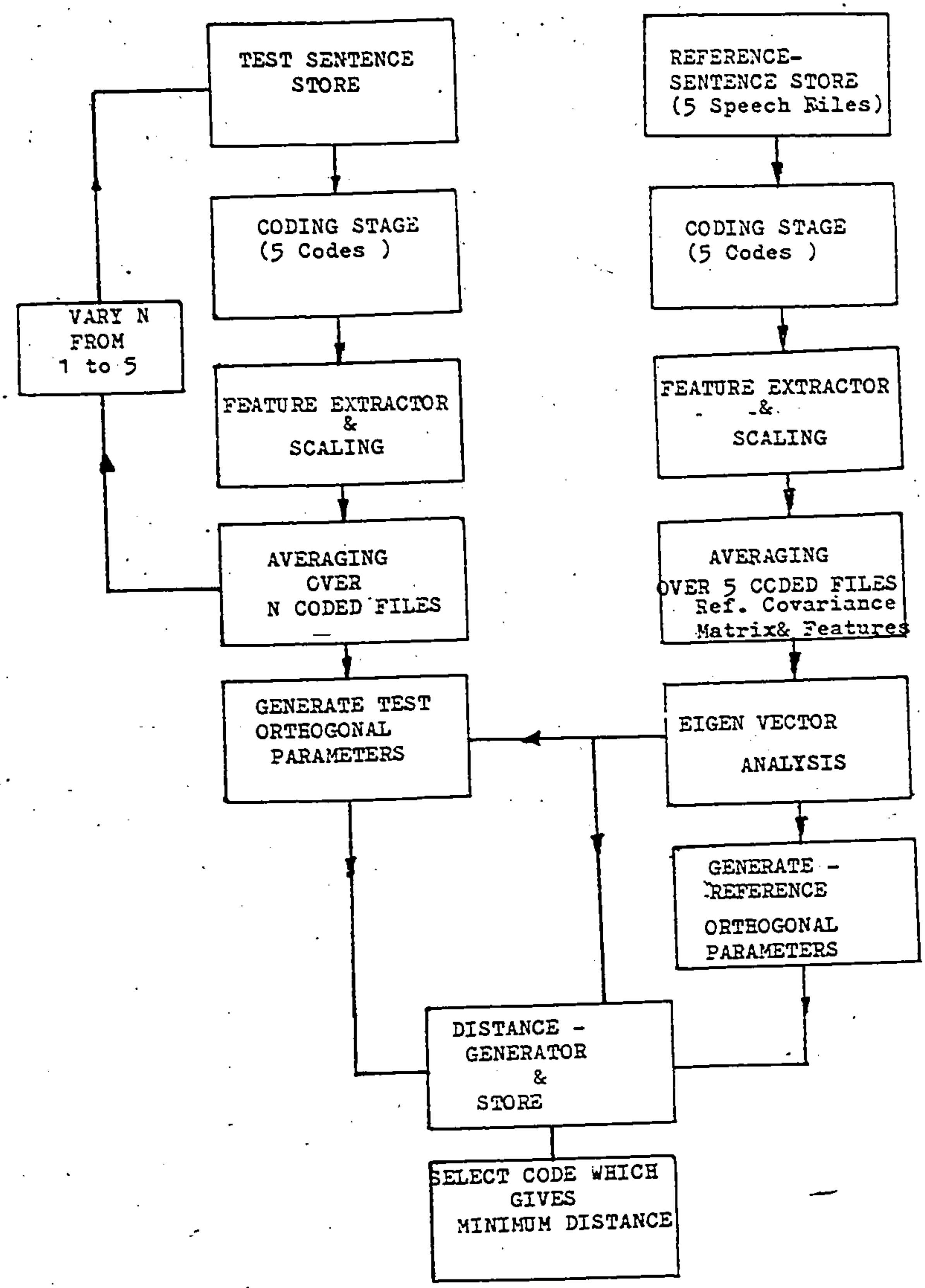


Fig. 5.3 Flowchart of Algorithm # 1

5.2.2 The Testing Phase

The Testing Phase consists of the same operations as in the Training Stage. The main difference between the two phases of operation is that the speech data for the testing stage is derived from the Test Sentence Store.

Similiar operations of Coding, Feature Extraction and Scaling as in the previous section are carried out.

The Averaging Operation is the same in all respects to the process carried out in the training phase with regard to the number of test files, N, used for generating the test feature set. Here, the number of test files, N, used for the averaging process is not fixed at five, but is made variable from one to five.

5.2.3 The Classification Phase

In the Classification phase of the algorithm, the Reference Covariance Matrix for each code is constructed from the Reference Feature Set for each code and an Eigenvector Analysis (Section 4.3.2) is carried out on each of these covariance matrices, to derive Reference Eigenvectors and Reference Eigenvalues.

Using these Reference Eigenvectors, Reference Orthogonal Parameters are derived by taking the dot product of the eigenvectors and the reference feature vectors.

Similiarly, Test Orthogonal Parameters are generated by taking the dot product of the Reference Eigenvectors and the TestAveraged-Feature Vectors.

The final Classification is carried out in the Orthogonal Domain by computing the Orthogonal Distance Metric given by:

$$d_{ik} = \sum_{j=1}^{NPT} \frac{(\phi_{ji} - \alpha_{jk})^2}{\lambda_i}$$

$$i = 1, 2, \text{---}, 5$$
$$k = 1, 2, \text{---}, 5$$

where,

d_{ik} is the distance computed between the i th reference code and k th test code.

ϕ_{ji} is the j th reference orthogonal parameter of the i th reference code.

α_{jk} is the k th test orthogonal parameter of the k th test code.

λ_i is the reference eigen value of the i th reference code.

NPT is the number of Orthogonal Parameters used in the distance computation.

The orthogonal distance measure is computed between all the test code patterns and all the reference code patterns, and the minimum distance computed between each test code and reference code pattern is stored and the reference code corresponding to that minimum distance is noted.

Thus the number of misidentifications can be found and noted as the number of test files used in generating the test feature set is varied from one to five. In addition, the number of orthogonal parameters being used in the distance metric is reduced to two, so that only the first two most-effective parameters are used in the distance computation, and the number of misidentifications is noted. Figure 5.3(a) gives the results of the identification of the codes when all three features are used in the distance computation. It can be seen that the accuracy of identification of the system decreases, as the number of test files is increased.

Figure 5.3(b) gives the results of the identification accuracy when only the first two features are used in the distance computation. Here it can be seen that significant improvements in the Code-Identification results as the number of test speech files is increased. When the number of test files used reaches five, all the codes are identified correctly, and the percentage of misidentification is zero.

Notations

PAM : Pulse Amplitude Modulation

DPAM-F : First Order Fixed Predictor Differential Pulse Amplitude Modulation

DPAM-1 : First Order Adaptive Predictor Differential Pulse Amplitude Modulation

DPAM-2 : Second Order Adaptive Predictor Differential Pulse Amplitude Modulation

LOG-PAM : Log-Pulse Amplitude Modulation (U=[illegible]

NSpF : No. of Test Speech Files

NMI : No. of Misidentifications

CODES AS IDENTIFIED BY THE SYSTEM		CODES USED FOR TESTING THE SYSTEM (ACTUAL)					
	NSpF	PAM	DPAM-F	DPAM-1	DPAM-2	LOG-PAM	NMI
	1	PAM	DPAM-F	DPAM-1	PAM	LOG-PAM	1
	2	PAM	DPAM-F	DPAM-1	PAM	LOG-PAM	1
	3	PAM	DPAM-F	DPAM-1	PAM	LOG-PAM	2
	4	PAM	DBAM-F	DPAM-1	PAM	DPAM-1	2
	5	PAM	DPAM-F	DPAM-1	PAM	DPAM-1	2

Fig. 5.3(a) Results Obtained From Algorithm # 1 Using All Three Orthogonal Parameters

CODES AS IDENTIFIED BY THE SYSTEM		CODES USED FOR TESTING THE SYSTEM (ACTUAL)					
	NSpF	PAM	DPAM-F	DPAM-1	DPAM-2	LOG-PAM	NMI
	1	PAM	DPAM-F	DPAM-1	PAM	LOG-PAM	1
	2	PAM	DPAM-F	DPAM-1	PAM	LOG-PAM	1
	3	PAM	DPAM-F	DPAM-1	PAM	LOG-PAM	1
	4	PAM	DPAM-F	DPAM-1	PAM	LOG-PAM	1
	5	PAM	DPAM-F	DPAM-1	DPAM-2	LOG-PAM	0

Fig. 5.3(b) Results Obtained from Algorithm # 1 Using the First Two Orthogonal Parameters

This result gives an indication of the duration of the coded speech signal which is necessary for accurate identification of all the codes, using this algorithm (Algorithm #1). If five test files are required for generating the test orthogonal parameter set, then that means 8.0 seconds of coded speech signal is necessary for all the codes to be identified correctly.

However, if Second Order Adaptive Predictor DPAM is not included in the coding scheme set, then just 1.6 seconds of coded speech signal is sufficient for accurate identification, using the Algorithm #1.

Chapter VI

CONCLUSIONS

The purpose of this thesis was to carry out an investigation into Code Identification in a Digitally Coded Speech Signal using the techniques of Pattern Classification.

Five coding schemes were selected and implemented in software (Fortran-V) on the Data General Nova 840 Computer. These coding schemes were applied to processed speech sentences, from which reference and test signals were generated.

Short term features including Energy Contour, Zero Crossing Rate and Normalized Autocorrelation with Lag One were extracted from these signals, and were suitably transformed and normalized to form Reference and Test Patterns for each of the codes.

The Classification of the Test Patterns was carried out using the algorithm which was developed above.

The algorithm for code-identification used the Orthogonal Distance Measure for the Classification, and the results that were obtained by applying this algorithm are tabulated.

The main problem encountered in this study was the problem of making the features Sentence Independent.

To tackle this problem Long Term Averaging was used on the features. Two different kinds of Averaging Schemes were tried out on the transformed features, and the effects of these schemes were observed using scatter plots. The accuracy of Identification was found to depend on the length of the Coded Speech Signal from which the features were derived for the averaging.

The third orthogonal parameter derived from the raw features was found to reduce the accuracy of identification rather than increase or not affect the accuracy of identification. This was indicated earlier by the scatter plots too. Thus the first two orthogonal parameters are the most effective in code identification.

Considering the results of applying Algorithm #1, it is found that the minimum length of coded signal required for correct identification of all the codes is 8.0 seconds.

However, if the Second Order Adaptive Predictor DPAM code is not included in the Code set under study, then the minimum length of coded speech signal required for correct identification reduces to 1.6 seconds.

In conclusion, this work shows that Code-Identification in a Digitally Coded Signal is feasible.

6.1 SUGGESTIONS FOR FURTHER WORK

For further investigation into this topic, a couple of suggestions about the areas in which more research could be carried out are made here.

(1) Using other raw features like LPC's and frequency domain parameters, more efficient transformed parameters for Code Identification may be generated.

This could result in the reduction of the minimum length of coded signal required for the averaging and classification.

(2) An hierarchical type of Code Identification System could be investigated wherein, the first stage of identification could be the identifying of the General classes of Coding Schemes, such as Uniform Quantization, Non-Uniform Quantization and Differential Coding Schemes. The next level in the system could be the identification of the actual codes.

This system could make the identification scheme more general in application.

APPENDIX- A

DERIVATION OF PREDICTION COEFFICIENTS α_1 & α_2 FOR THE DIFFERENTIAL QUANTIZATION SCHEMES

The Linear Prediction System with Prediction Coefficients, α_k's, is defined by the equation:

$$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k s(n-k)$$

The Prediction Error , e(n), is defined as:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k)$$

The prediction coefficients, α_k's are derived such that the mean squared error over a short segment of the speech waveform $s_n(m)$ is minimised. The short-time averaged prediction error is defined as:

$$E = e_n^2(m)$$

$$= (s(m) - \tilde{s}(m))^2 = \sum_m \left[s_n(m) - \sum_{k=1}^{p} \alpha_k s_n(m-k) \right]^2$$

where, the range of the summation is a finite number temporarily left unspecified, and $s_n(m)$ is a segment of speech in the vicinity of sample, n, ie. $s_n(m) = s(m+n)$.

Setting $E_n \quad i = 0$, i=1,2,---,p, the following equations are obtained:

$$\sum_m s_n(m - i) s_n(m) = \sum_{k=1}^{p} \alpha_k \sum_m s_n(m-i) s_n(m-k) \qquad 1 \leqslant i \leqslant p \quad \text{---(1)}$$

Let $\phi_n(i,k) = \sum_m s_n(m-i)s_n(m-k)$ ------------------------(2)

Then,

$$\sum_{k=1}^{p} \alpha_k \phi_n(i,k) = \phi_n(i,0) \qquad i=1,2,\text{------},p$$

These equations can be solved by assuming that the waveform segment, $s_n(m)$, is zero outside the interval $0 \leq m \leq N-1$.

The short time average error E_n can be expressed by :

$$E_n = \sum_{m=0}^{N+P-1} e_n^2(m)$$

Equation (2) can then be written as :

$$\phi_n(i,k) = \sum_{m=0}^{N+P-1} s_n(m-i)s_n(m-k) \qquad \begin{matrix} 1 \leq i \leq p \\ 0 \leq i \leq p \end{matrix}$$

$$= \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k) \qquad \begin{matrix} 1 \leq i \leq p \\ 0 \leq k \leq p \end{matrix}$$

In this case, $\phi_n(i,k)$ is identical to the short-time autocorrelation function $R_n(i-k)$. That is,

$$\phi_n(i,k) = R_n(i-k)$$

where,

$$R_n(k) = \sum_{m=0}^{N-1-k} s_n(m)s_n(m+k)$$

Since $R_n(k)$ is an even function,

$$\phi_n(i,k) = R_n(\ i-k\ ) \qquad \begin{matrix} i = 1,2,\text{-----},p \\ k = 0,1,\text{-----},p \end{matrix}$$

Equation (3) can therefore be expressed as

$$\sum_{k=1}^{p} \alpha_k R_n(i-k) = R_n(i) \qquad 1 \leq i \leq p \quad \text{-----------} (4)$$

Equations (4) can be expressed in matrix form as

$$\begin{bmatrix} R_n(0) & R_n(1) & R_n(2) & \cdots & R_n(p-1) \\ R_n(1) & R_n(0) & R_n(1) & \cdots & R_n(p-2) \\ R_n(2) & R_n(1) & R_n(0) & \cdots & R_n(p-3) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ R_n(p-1) & R_n(p-2) & R_n(p-3) & \cdots & R_n(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \cdots \\ \cdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ R_n(3) \\ \cdots \\ \cdots \\ R_n(p) \end{bmatrix} \text{--} (5)$$

FIRST ORDER ADAPTIVE PREDICTIVE DPAM

For the First Order Adaptive Predictor DPAM coding scheme, the predictor coefficient α_1 is found by substituting p=1 in equations (4) or (5).

We thus get the equation for α_1 as

$$\alpha_1 = \frac{R(1)}{R(0)}$$

SECOND ORDER ADAPTIVE PREDICTOR DPAM

For the Second Order Adaptive Predictor DPAM coding scheme, the predictor coefficients α_1 and α_2 are computed by substituting p = 2 in equations (4) or (5). Two simultaneous equations are obtained, which on solving for α_1 and α_2, give the following expressions:

$$\alpha_1 = \frac{R(0)\,R(1) - R(1)\,R(2)}{R^2(0) - R^2(1)}$$

$$\alpha_2 = \frac{R^2(1) - R(2)\,R(0)}{R^2(1) - R^2(0)}$$

APPENDIX-B

LISTING OF COMPUTER PROGRAMS

```
C         ************************************************
C
C         THIS PROGRAM CARRIES OUT THE OPERATIONS OF CODING
C         AND FEATURE EXTRACTION AS DESCRIBED BY THE
C         ALGORITHM.
C
C         THIS PROGRAM GENERATES AN OUTPUT FILE (NAME TO BE
C         SPECIFIED BY THE USER) CONTAINING THE THREE
C         FEATURES- ENERGY CONTOUR, NORMALISED AUTOCORRELATION
C         WITH LAG ONE AND ZERO CROSSING RATE, WHICH ARE
C         EXTRACTED FROM THE CODED SPEECH SIGNALS GENERATED
C         BY CODING EACH OF THE INPUT SPEECH FILES WITH
C         EACH OF THE FIVE CODES.
C
C         THE INFORMATION TO BE SUPPLIED BY THE USER IS:
C
C           (1) THE OUTPUT-FILE NAME
C           (2) THE INPUT-SPEECH-FILE NAME
C           (3) WHETHER THE USER WISHES TO OPEN A NEW
C               OUTPUT FILE OR TO APPEND TO AN EXISTING ONE.
C
C         ************************************************
C
        DIMENSION ID(200),IP(200),IDHAT(200),SIGX(200)
        DIMENSION IXL(200),ECA(80),ACA(80),ZCA(80),ECB(80)
        DIMENSION ACB(80),ZCB(80),ECC(80),ACC(80),ZCC(80)
        DIMENSION ACE(80),ZCE(80),NAME(6)
        DIMENSION ECD(80),ACD(80),ZCD(80),ECE(80)
        DIMENSION NAME1(16)
        TYPE 'TYPE IN OUTPUT FILENAME'
        READ(11,2) (NAME1(I),I=1,8)
2       FORMAT(8A2)
        TYPE 'WISH TO OPEN OR APPEND THE OUTPUT FILE ? '
        ACCEPT '(OPEN=1 AND APPEND=0 )',IWO
        IF(IWO.NE.1) GO TO 23
        OPEN 1,NAME1
        GO TO 788
23      CALL APPEND(1,NAME1,DUMMY,IER)
        CALL CHECK(IER)
        GO TO 788
788     IWISH=1
888     ACCEPT 'WISH TO CONTINUE ?' ,IWISH
        IF(IWISH.NE.1) GO TO 999
        TYPE "ENTER INPUT FILENAME"
        READ(11,1) (NAME(I),I=1,3)
1       FORMAT(3A2)
        TYPE 'OK1'
        OPEN 6,NAME
        TYPE 'OK2'
        XMAX=-9999999.
        DO 90 K=1,80
        READ (6) (ID(I),I=1,200)
        DO 80 J=1,200
        XMAX=AMAX1(XMAX,ID(J))
80      CONTINUE
```

```
90      CONTINUE
        CLOSE (6)
        OPEN 6,NAME
        DO 99 KOUNT=1,80
        READ (6) (ID(I),I=1,200)
        CALL PAM(ID,200,IP)
        CALL FEXTR2(IP,200,EC1,ZC1,AC1)
        ECA(KOUNT)=EC1
        ACA(KOUNT)=AC1
        ZCA(KOUNT)=ZC1
        CALL PREDCF(ID,200,AF1,AS1,AS2)
        CALL DPAMF(KOUNT,ID,IDHAT,1.,XTLF)
        CALL FEXTR2(IDHAT,200,EC1,ZC1,AC1)
        ECB(KOUNT)=EC1
        ACB(KOUNT)=AC1
        ZCB(KOUNT)=ZC1
        CALL DPAM1(KOUNT,ID,IDHAT,AF1,XTL)
        CALL FEXTR2(IDHAT,200,EC1,ZC1,AC1)
        ECC(KOUNT)=EC1
        ACC(KOUNT)=AC1
        ZCC(KOUNT)=ZC1
        CALL DPAM2(KOUNT,ID,IDHAT,AS1,AS2,XTL2,IXHT)
        CALL FEXTR2(IDHAT,200,EC1,ZC1,AC1)
        ECD(KOUNT)=EC1
        ACD(KOUNT)=AC1
        ZCD(KOUNT)=ZC1
        U=100.
        DO 161 I=1,200
        IF(ID(I).GE.0) GO TO 21
        SIGX(I)=-1.
        GO TO 162
21      SIGX(I)=1.
162     IXL(I)=(XMAX/ALOG10(1+U))*ALOG10(1.
       &+100.+ABS(ID(I))/XMAX))*SIGX(I)
161     CONTINUE
        CALL PAM(IXL,200,IP)
        CALL FEXTR2(IP,200,EC1,ZC1,AC1)
        ECE(KOUNT)=EC1
        ACE(KOUNT)=AC1
        ZCE(KOUNT)=ZC1
99      CONTINUE
        DO 20 I=1,80
        WRITE(1,33) ECA(I),ACA(I),ZCA(I)
        WRITE(1,33) ECB(I),ACB(I),ZCB(I)
        WRITE(1,33) ECC(I),ACC(I),ZCC(I)
        WRITE(1,33) ECD(I),ACD(I),ZCD(I)
        WRITE(1,33) ECE(I),ACE(I),ZCE(I)
33      FORMAT(1X,E15.7,1X,E15.7,1X,E15.7)
20      CONTINUE
        CLOSE (6)
        GO TO 888
999     CONTINUE
        CLOSE 1
        STOP
```

```
C      *************************************************
C      ***                                           ***
C      ***  SUBROUTINE PAM(IX,NSF,IP)                ***
C      ***                                           ***
C      *************************************************
C
C      THE SUBROUTINE CONVERTS THE INPUT SPEECH
C      SIGNAL SEGMENT, IX, INTO A PULSE AMPLITUDE MODULATED
C      OUTPUT SIGNAL, IP.
C      THE FRAME SIZE OF EACH SEGMENT IS DETERMINED BY NSF
C
C
      SUBROUTINE PAM(IX,NSF,IP)
      DIMENSION IX(NSF),IP(NSF)
      XMAX=.3E 04
      XMIN=-.3E 04
      A=255./(XMAX-XMIN)
      B=127.-A*XMAX
      DO 10 JK=1,NSF
      IP(JK)=A*FLOAT(IX(JK))+B
10    CONTINUE
      RETURN
```

```
C     **************************************************
C     ***                                            ***
C     ***   SUBROUTINE PREDCF(IX,NSF,AF1,AS1,AS2)    ***
C     ***                                            ***
C     **************************************************
C
C     THIS SUBROUTINE COMPUTES THE FORST AND SECOND ORDER
C     PREDICTOR COEFFICIENTS FROM THE INPUT SPEECH SEGMENT,
C     IX.
C
C     NSF DETERMINES THE FRAME SIZE OF EACH SEGMENT.
C
C     AF1 IS THE FIRST ORDER PREDICTOR FROM SEGMENT, IX.
C
C     AS1 IS THE FIRST COEFFICIENT IN THE SECOND ORDER
C     PREDICTION SYSTEM.
C
C     AS2 IS THE SECOND COEFFICIENT IN THE SECOND ORDER
C     PREDICTION SYSTEM.
C
C
      DIMENSION IX(NSF),R(30)
      DO 12 K=1,4
      R(K)=0.0
      NK=NSF-K+1
      DO 12 NP=1,NK
      R(K)=R(K)+FLOAT(IX(NP)*IX(NP+K-1))
12    CONTINUE
      AF1=R(2)/R(1)
      AS1=(R(1)*R(2)-R(2)*R(3))/(R(1)*R(1)-R(2)*R(2))
      AS2=(R(2)*R(2)-R(3)*R(1))/(R(2)*R(2)-R(1)*R(1))
      RETURN
```

```
C       ***********************************************
C       ***                                         ***
C       ***  SUBROUTINE DPAMF(KOUNT,IX,IDHAT,A1,XTLF)  ***
C       ***                                         ***
C       ***********************************************
C
C       THIS SUBROUTINE CONVERTS THE INPUT SPEECH SIGNAL
C       SEGMENT,IX, INTO A FIRST ORDER FIXED PREDICTOR
C       DPAM SIGNAL, IDHAT.
C       A1 DETERMINES THE FIRST ORDER FIXED PREDICTOR USED
C       FOR THE CODING.
C       KOUNT IS THE PARTICULAR SEGMENT OF THE SPEECH FILE
C       THAT IS BEING CODED.
C
      SUBROUTINE DPAMF(KOUNT,IX,IDHAT,A1,XTLF)
      DIMENSION IX(200),XTILD(210),D(200),IDHAT(200),IXHAT(210)
      XMAX=.3E 04
      XMIN=-.3E 04
      A=255./(XMAX-XMIN)
      B=127.-A*XMAX
      IF(KOUNT.NE.1) GO TO 99
      XTILD(1)=0.0
      D(1)=FLOAT(IX(1)-IFIX(XTILD(1)))
      IDHAT(1)=IFIX(A*(D(1))+B)
      IXHAT(1)=IDHAT(1)+IFIX(XTILD(1))
      DO 1 I=2,200
      XTILD(I)=A1*FLOAT(IXHAT(I-1))
      D(I)=FLOAT(IX(I)-IFIX(XTILD(I)))
      IDHAT(I)=IFIX(A*D(I)+B)
      IXHAT(I)=IDHAT(I)+IFIX(XTILD(I))
1     CONTINUE
      XTLF=XTILD(200)
      RETURN
99    XTILD(1)=XTLF
      DO 2 I=1,200
      D(I)=FLOAT(IX(I)-IFIX(XTILD(I)))
      IDHAT(I)=IFIX(A*D(I)+B)
      IXHAT(I)=IDHAT(I)+IFIX(XTILD(I))
      XTILD(I+1)=A1*FLOAT(IXHAT(I))
2     CONTINUE
      XTLF=XTILD(200)
      RETURN
```

```
C     ***********************************************
C     ***                                         ***
C     ***  SUBROUTINE DPAM1(KOUNT,IX,IDHAT,A1,XTL ***
C     ***                                         ***
C     ***********************************************
C
C     THIS SUBROUTINE CONVERTS THE INPUT SPEECH SIGNAL
C     SEGMENT, IX, INTO A FIRST ORDER ADAPTIVE DPAM SIGNAL
C     IDHAT.
C     THE VALUE OF THE FIRST ORDER ADAPTIVE PREDICTOR
C     COMPUTED OVER THE SEGMENT IS FED TO THE SUBROUTINE
C     THROUGH A1.
C     KOUNT IS THE PARTICULAR SEGMENT OF THE SPEECH FILE
C     THAT IS BEING CODED.
C
C
      SUBROUTINE DPAM1(KOUNT,IX,IDHAT,A1,XTL)
      DIMENSION IX(200),XTILD(210),D(200),IDHAT(200),IXHAT(210)
      XMAX=.3E 04
      XMIN=-.3E 04
      A=255./(XMAX-XMIN)
      B=127.-A*XMAX
      IF(KOUNT.NE.1) GO TO 99
      XTILD(1)=0.0
      D(1)=FLOAT(IX(1)-IFIX(XTILD(1)))
      IDHAT(1)=IFIX(A*(D(1))+B)
      IXHAT(1)=IDHAT(1)+IFIX(XTILD(1))
      DO 1 I=2,200
      XTILD(I)=A1*FLOAT(IXHAT(I-1))
      D(I)=FLOAT(IX(I)-IFIX(XTILD(I)))
      IDHAT(I)=IFIX(A*D(I)+B)
      IXHAT(I)=IDHAT(I)+IFIX(XTILD(I))
1     CONTINUE
      XTL=XTILD(200)
      RETURN
99    XTILD(1)=XTL
      DO 2 I=1,200
      D(I)=FLOAT(IX(I)-IFIX(XTILD(I)))
      IDHAT(I)=IFIX(A*D(I)+B)
      IXHAT(I)=IDHAT(I)+IFIX(XTILD(I))
      XTILD(I+1)=A1*FLOAT(IXHAT(I))
2     CONTINUE
      XTL=XTILD(200)
      RETURN
```

```
C     ************************************************************
C     ***                                                      ***
C     ***  SUBROUTINE DPAM2(KOUNT,IX,IDHAT,A1,A2,XTL2,IXHT)    ***
C     ***                                                      ***
C     ************************************************************
C
C      THIS SUBROUTINE CONVERTS THE INPUT SPEECH SIGNAL SEGMENT
C      IX, INTO A SECOND ORDER ADAPTIVE PREDICTOR DPAM SIGNAL,
C      IDHAT.
C      THE VALUES OF THE FIRST AND SECOND ORDER ADAPTIVE
C      PREDICTORS COMPUTED OVER THE SPEECH SEGMENT BEING CODED
C      ARE FED TO THE SUBROUTINE THROUGH A1 AAND A2 RESPECTIVELY.
C      KOUNT IS THE PARTICULAR SEGMENT OF THE SPEECH FILE BEING
C      CODED.
C
C
      SUBROUTINE DPAM2(KOUNT,IX,IDHAT,A1,A2,XTL2,IXHT)
      DIMENSION IX(200),XTILD(210),D(200),IDHAT(200),IXHAT(210)
      XMAX=.3E 04
      XMIN=-.3E 04
      A=255./(XMAX-XMIN)
      B=127.-A*XMAX
      IF(KOUNT.NE.1) GO TO 99
      XTILD(1)=0.0
      D(1)=FLOAT(IX(1)-IFIX(XTILD(1)))
      IDHAT(1)=A*FLOAT(D(1))+B
      IXHAT(1)=IDHAT(1)+IFIX(XTILD(1))
      XTILD(2)=A1*FLOAT(IXHAT(1))
      D(2)=FLOAT(IX(2)-IFIX(XTILD(2)))
      IDHAT(2)=A*FLOAT(D(I))+B
      IXHAT(2)=IDHAT(2)+IFIX(XTILD(2))
      DO 1 I=3,200
      XTILD(I)=A1*FLOAT(IXHAT(I-1))+A2*FLOAT(IXHAT(I-2))
      D(I)=FLOAT(IX(I)-IFIX(XTILD(I)))
      IDHAT(I)=A*FLOAT(D(I))+B
      IXHAT(I)=IDHAT(I)+IFIX(XTILD(I))
1     CONTINUE
      XTL2=XTILD(200)
      IXHT=IXHAT(200)
      RETURN
99    XTILD(1)=XTL2
      IXHAT(1)=IXHT
      DO 2 I=1,200
      D(I)=FLOAT(IX(I)-IFIX(XTILD(I)))
      IDHAT(I)=A*FLOAT(D(I))+B
      IXHAT(I+1)=IDHAT(I)+IFIX(XTILD(I))
      XTILD(I+1)=A1*FLOAT(IXHAT(I+1))+A2*FLOAT(IXHAT(I))
2     CONTINUE
      XTL2=XTILD(200)
      IXHT=IXHAT(200)
      RETURN
```

```
C       ************************************************
C       ***                                          ***
C       ***   SUBROUTINE FEXTR2(IX,NSF,E1,ZCR,AC1)   ***
C       ***                                          ***
C       ************************************************
C
C       THIS SUBROUTINE EXTRACTS THE THREE FEATURES,
C       ENERGY CONTOUR, NORMALISED AUTOCORRELATION WITH
C       LAG ONE, AND ZERO CROSSING RATE FOM THE INPUT
C       CODED SPEECH SEGMENT, IX.
C
C       NSF DETERMINES THE FRAME SIZE OF THE SIGNAL SEGMENT.
C
C       E1 IS THE SHORT TERM ENERGY OVER THE CODED SIGNAL
C       SEGMENT.
C
C       ZCR IS THE SHORT TERM ZERO CROSSING RATE OF THE
C       CODED SIGNAL SEGMENT.
C
C       AC1 IS THE SHORT TERM (NORMALISED) AUTOCORRELATION
C       WITH LAG ONE OVER THE CODED SIGNAL SEGMENT.
C
      SUBROUTINE FEXTR2(IX,NSF,E1,ZCR,AC1)
      DIMENSION IX(NSF),R(30),SIGX(300),RC(20),A(20),RF(30)
C      COMPUTE SHORT TERM ENERGY POINT
      NORM=1
      SUME=0.
      DO 11 I=1,NSF
      SUME=SUME+FLOAT(IX(I))**2
11    CONTINUE
      E1=SUME/(FLOAT(NSF*1000000))
C      COMPUTE AUTOCORRELATION WITH LAG ONE
      DO 12 K=1,4
      R(K)=0.
      NK=NSF-K+1
      DO 12 NP=1,NK
      R(K)=R(K)+FLOAT(IX(NP))*FLOAT(IX(NP+K-1))
12    CONTINUE
      IF(NORM.NE.1) GO TO 111
      ENRG=R(1)
      DO 13 I=1,4
      RF(I)=R(I)/ENRG
13    CONTINUE
      AC1=RF(2)
      GO TO 65
111      DO 60 I=1,4
      RF(I)=R(I)
60    CONTINUE
C      COMPUTE ZERO CROSSING RATE FOR EACH SEGMENT
65    DO 41 I=1,NSF
      IF(IX(I).GE.0) GO TO 21
      SIGX(I)=-1.
      GO TO 41
21    SIGX(I)=1.
41    CONTINUE
```

```
      SUMZ=0.
      ND=NSF-1
      DO 51 I=1,ND
      SUMZ=SUMZ+ABS(SIGX(I)-SIGX(I+1))
51    CONTINUE
      ZCR=SUMZ/(FLOAT(NSF)*2.)
      RETURN
```

```
C       **************************************************
C
C       THIS PROGRAM CARRIES OUT THE OPERATIONS OF AVERAGING
C       THE FEATURES OVER MANY SENTENCES AND CLASSIFICATION
C       AS DESCRIBED IN THE ALGORITHM.
C       THE REFERENCE AND TEST FILES GENERATED BY THE PREVIOUS
C       PROGRAM ARE USED BY THIS PROGRAM FOR AVERAGING AND
C       CLASSIFICATION.
C
C       THE NAMES OF THE REFERENCE AND TEST FILES (CONTAINING
C       THE FEATURES) IS TO BE SPECIFIED BY THE USER.
C
C       THE NUMBER OF SPEECH FILES OVER WHICH THE AVERAGING
C       IS TO BE CARRIED OUT TO GENERATE THE REFERENCE AND
C       TEST FEATURE VECTORS FOR THE CLASSIFICATION IS ALSO
C       TO BE SPECIFIED BY THE USER.
C
C       **************************************************
C
C
        DIMENSION ECA(160),ECB(160),ECC(160),ECD(160)
        DIMENSION ECE(160),ACA(160),ACB(160)
        DIMENSION ACC(160),ACD(160),ACE(160),ZCA(160)
        DIMENSION ZCB(160),ZCC(160),ZCD(160)
        DIMENSION ZCE(160),CX(5,3,3)
        DIMENSION C(300,3),SUM(3,5),CR(3,3)
        DIMENSION XR(5,3,5),NAME1(16),NAME2(16),XT(5,3,5)
        DIMENSION TX(3,5),RX(3,5),EIG(9),PHIREF(3,5)
        DIMENSION XT(5,3,5),PHIX(3,5),DST(5)
        OPEN 3,'CVMX1'
        TYPE 'TYPE REFERENCE FILENAME'
        READ(11,12) (NAME2(I),I=1,8)
        TYPE 'TYPE IN TEST SIGNAL FILENAME'
        READ(11,12) (NAME1(I),I=1,8)
12      FORMAT(8A2)
        KOUNT=0
981     ACCEPT 'WISH TO CONTINUE ',IWISH
        IF(IWISH.NE.1) GO TO 999
        IF(KOUNT.NE.0) GO TO 888
        OPEN 1,NAME2
        GO TO 887
888     OPEN 1,NAME1
887     ACCEPT 'NO. OF SPEECH FILES',NFILES
        DO 2 I=1,3
        DO 2 J=1,5
        SUM(I,J)=0.0
2       CONTINUE
        DO 122 IT=1,NFILES
        DO 77 I=1,80
        READ (1,33) ECA(I),ACA(I),ZCA(I)
        READ (1,33) ECB(I),ACB(I),ZCB(I)
        READ (1,33) ECC(I),ACC(I),ZCC(I)
        READ (1,33) ECD(I),ACD(I),ZCD(I)
        READ (1,33) ECE(I),ACE(I),ZCE(I)
33      FORMAT(1X,E15.7,1X,E15.7,1X,E15.7)
```

```
77    CONTINUE
      DO 3 I=1,80
      SUM(1,1)=SUM(1,1)+ECA(I)
      SUM(2,1)=SUM(2,1)+ACA(I)
      SUM(3,1)=SUM(3,1)+ZCA(I)
      SUM(1,2)=SUM(1,2)+ECB(I)
      SUM(2,2)=SUM(2,2)+ACB(I)
      SUM(3,2)=SUM(3,2)+ZCB(I)
      SUM(1,3)=SUM(1,3)+ECC(I)
      SUM(2,3)=SUM(2,3)+ACC(I)
      SUM(3,3)=SUM(3,3)+ZCC(I)
      SUM(1,4)=SUM(1,4)+ECD(I)
      SUM(2,4)=SUM(2,4)+ACD(I)
      SUM(3,4)=SUM(3,4)+ZCD(I)
      SUM(1,5)=SUM(1,5)+ECE(I)
      SUM(2,5)=SUM(2,5)+ACE(I)
      SUM(3,5)=SUM(3,5)+ZCE(I)
3     CONTINUE
      IF(KOUNT.NE.0) GO TO 122
      CALL COVAR(ECA,ACA,ZCA,3,80,80)
      CALL COVAR(ECB,ACB,ZCB,3,80,80)
      CALL COVAR(ECC,ACC,ZCC,3,80,80)
      CALL COVAR(ECD,ACD,ZCD,3,80,80)
      CALL COVAR(ECE,ACE,ZCE,3,80,80)
122   CONTINUE
      CALL RESET
      DO 44 I=1,3
      DO 44 J=1,5
      SUM(I,J)=SUM(I,J)/FLOAT(NFILES*80)
      WRITE(10,33) SUM(I,J)
44    CONTINUE
      TYPE 'KOUNT=',KOUNT
      IF(KOUNT.NE.0) GO TO 41
      OPEN 3,'CVMX1'
      NTI=3*5*NFILES
      DO 212 I=1,NTI
      READ(3,33) (C(I,J),J=1,3)
212   CONTINUE
      CLOSE 3
      DO 202 L=1,5
      DO 202 J=1,3
      DO 202 I=1,3
      CX(L,I,J)=0.0
202   CONTINUE
      ICODE=0
      DO 203 L=1,5
      DO 204 J=1,3
      DO 205 I=1,3
      IADD=0
      DO 206 K=1,NFILES
      CX(L,I,J)=CX(L,I,J)+C(I+IADD+ICODE,J)
      IADD=IADD+15
206   CONTINUE
205   CONTINUE
204   CONTINUE
```

```
      ICODE=ICODE+3
203   CONTINUE
      DO 207 L=1,5
      DO 207 J=1,3
      DO 207 I=1,3
      CX(L,I,J)=CX(L,I,J)/FLOAT(NFILES)
207   CONTINUE
      WRITE(10,130) NFILES
      WRITE(12,130) NFILES
130   FORMAT(15X,'NO. OF FILES FOR REF. TEMPLATES :-',I3,////)
      DO 6 I=1,3
      DO 6 J=1,5
      XR(J,I,1)=SUM(I,J)
      WRITE(12,59) XR(J,I,1),SUM(I,J)
      WRITE(10,59) XR(J,I,1),SUM(I,J)
6     CONTINUE
      IF(KOUNT.NE.0) GO TO 41
      KOUNT=KOUNT+1
      GO TO 991
41    DO 7 I=1,3
      DO 7 J=1,5
      XT(J,I,1)=SUM(I,J)
      WRITE(10,59) XR(J,I,1),XT(J,I,1)
59    FORMAT(13X,E15.7,13X,E15.7)
7     CONTINUE
      WRITE(10,800) NFILES
      WRITE(12,800) NFILES
800   FORMAT(10X,'NO. OF FILES FOR TEST SIGNAL:-',I3)
      DO 111 I=1,5
      DO 112 J=1,3
      DO 113 K=1,1
      TX(J,K)=XT(I,J,K)
113   CONTINUE
112   CONTINUE
      DO 114 L=1,5
      DO 445 IC=1,3
      DO 445 JC=1,3
      CR(IC,JC)=CX(L,IC,JC)
445   CONTINUE
      DO 115 M=1,3
      DO 116 N=1,1
      RX(M,N)=XR(L,M,N)
      WRITE(10,33) RX(M,N),TX(M,N)
116   CONTINUE
115   CONTINUE
      CALL ORG(CR,RX,TX,EIG,PHIREF,PHIX,3,1)
      CALL DIST(PHIREF,PHIX,EIG,3,1,DRX)
      DST(L)=DRX
114   CONTINUE
      CALL CODNAM(I,IPOS,1)
      DO 71 KL=1,5
      WRITE(10,33) DST(KL)
      WRITE(12,33) DST(KL)
71    CONTINUE
      SMALL=DST(1)
```

```
      IPOS=1
      DO 100 KJ=2,5
      IF(SMALL.LT.DST(KJ)) GO TO 100
      IPOS=KJ
      SMALL=DST(KJ)
100   CONTINUE
      CALL CODNAM(I,IPOS,0)
      WRITE(10,150) SMALL
      WRITE(12,150) SMALL
150   FORMAT(5X,'MINIMUM DISTANCE COMPUTED =',E15.7,///)
111   CONTINUE
      GO TO 991
999   CONTINUE
      STOP
```

```
C      *************************************************
C      ***                                           ***
C      ***  SUBROUTINE COVAR(EC,AC,ZC,NFT,NPTS,NTOT)  ***
C      ***                                           ***
C      *************************************************
C
C      THIS SUBROUTINE COMPUTES THE COVARIANCE MATRIX
C      OVER THE FEATURES, NAMELY, ENERGY CONTOUR,NORMALISED
C      AUTOCORRELATION WITH LAG ONE AND ZERO CROSSING
C      RATE CONTOUR.
C
C      EC IS THE ENERGY CONTOUR DERIVED OVER THE CODED SPEECH
C      SIGNAL.
C
C      AC IS THE NORMALISED AUTOCORRELATION WITH LAG ONE DERIVED
C      OVER THE CODED SPEECH SIGNAL.
C
C      ZC IS THE ZERO CROSSING RATE DERIVED OVER THE CODED
C      SPEECH SIGNAL.
C
C      NPTS IS THE NUMBER OF FEATURE POINTS DERIVED FROM ONE
C      CODED SPEEECH SENTENCE (=80).
C
C      NTOT IS THE TOTAL NUMBER OF POINTS FOR EACH FEATURE
C      NTOT=80*NS, WHERE NS
C      IS THE TOTAL NUMBER OF SPEECH SENTENCES.
C
      SUBROUTINE COVAR(EC,AC,ZC,NFT,NPTS,NTOT)
      DIMENSION X(3,80),EC(80),AC(80),ZC(80)
      DIMENSION XM(3),C(3,3),XF(3,80)
      NP=NFT
      NA=NP*(NP+1)/2
      NR=NP*NP
      DO 1 J=1,NTOT
      XF(1,J)=EC(J)
      XF(2,J)=AC(J)
      XF(3,J)=ZC(J)
1     CONTINUE
      IADD=0
      DO 99 IKOUNT=1,NTOT,NPTS
      DO 4 I=1,NFT
      DO 4 J=1,NPTS
      X(I,J)=XF(I,J+IADD)
4     CONTINUE
99    CONTINUE
C      COMPUTE MEANS OF EACH FEATURE
      DO 2 I=1,NFT
      XM(I)=0.0
      DO 3 J=1,NPTS
      XM(I)=XM(I)+X(I,J)
3     CONTINUE
      XM(I)=XM(I)/FLOAT(NPTS)
2     CONTINUE
C      COMPUTE COVARIANCE MATRIX
      DO 10 L=1,NFT
```

```
      DO 11 M=1,NFT
      SUM=0.0
      DO 12 J=1,NPTS
      SUB1=X(L,J)-XM(L)
      SUB2=X(M,J)-XM(M)
      SUM=SUM+SUB1*SUB2
12    CONTINUE
      C(L,M)=SUM/FLOAT(NPTS-1)
11    CONTINUE
10    CONTINUE
      DO 13 L=1,NFT
      WRITE(3,33) (C(L,M),M=1,NFT)
33    FORMAT(1X,E15.7,1X,E15.7,1X,E15.7)
      WRITE(10,25) (C(L,M),M=1,NFT)
25    FORMAT(10X,E15.7,3X,E15.7,3X,E15.7)
13    CONTINUE
      RETURN
```

```
C       *********************************************************
C       *                                                       *
C       *SUBROUTINE ORG(C,X,TX,EIG,PHIREF,PHIX,NFT,NPTS)        *
C       *                                                       *
C       *********************************************************
C
C       THIS SUBROUTINE CARRIES OUT THE EIGENVECTOR
C       ANALYSIS ON THE REFERENCE COVARIANCE MATRIX
C       IN ORDER TO GENERATE REFERENCE EIGENVALUES.AND
C       REFERENCE EIGENVECTORS.
C       THE DOT PRODUCT OF THE REFERENCE-AVERAGED FEATURES
C       (PHIREF) WITH THE REFERENCE EIGENVECTORS TO GENERATE
C       REFERENCE ORTHOGONAL PARAMETERS AND THE DOT PRODUCT
C       OF THE TEST-AVERAGED FEATURES (PHIX) WITH THE
C       REFERENCE EIGENVECTORS TO GENERATE TEST ORTHOGONAL
C       PARAMETERS IS CARRIED OUT IN THIS SUBROUTINE.
C
C       C IS THE REFERENCE COVARIANCE MATRIX
C
C       PHIREF IS THE REFERENCE-AVERAGED FEATURE VECTOR
C
C       PHIX IS THE TEST-AVERAGED FEATURE VECTOR
C
C       NFT IS THE NUMBER OF FEATURES  (=3)
C
C       NPTS IS THE NUMBER OF POINTS PER FEATURE(=3)
C
C       EIG ARE THE REFERENCE EIGENVALUES COMPUTED BY
C       THE SUBROUTINE
C
      SUBROUTINE ORG(C,X,TX,EIG,PHIREF,PHIX,NFT,NPTS)
      DIMENSION X(3,5),TX(3,5),EIG(9)
      DIMENSION XM(3),C(3,3),A(6),E(3,3),R(9)
      DIMENSION PHIREF(3,5),PHIX(3,5)
      DO 133 I=1,3
      WRITE(10,33) X(I,1),TX(I,1)
33    FORMAT(10X,E15.7,10X,E15.7,10X,E15.7)
133   CONTINUE
      NP=NFT
      NA=NP*(NP+1)/2
      NR=NP*NP
      LK=1
      DO 60 K=1,NFT
      DO 60 I=1,K
      A(LK)=C(I,K)
      LK=LK+1
60    CONTINUE
      TYPE "OK 1"
      CALL EIGEN(A,R,NP,NA,NR,0)
      TYPE "OK 2"
      IX=1
      DO 85 IC=1,NFT
      DO 85 IR=1,NFT
      E(IR,IC)=R(IX)
      IX=IX+1
```

```
85    CONTINUE
C      IF EIGEN VALUES ARE REQUIRED
      L=1
      DO 80 K=1,NFT
      EIG(K)=A(L)
      L=L+K+1
80    CONTINUE
      DO 90 I=1,NFT
      DO 90 J=1,NPTS
      SUMPHX=0.0
      SUMPH=0.0
      DO 92 L=1,NFT
      SUMPH=SUMPH+E(L,I)*X(L,J)
      SUMPHX=SUMPHX+E(L,I)*TX(L,J)
92    CONTINUE
      PHIREF(I,J)=SUMPH
      PHIX(I,J)=SUMPHX
90    CONTINUE
      TYPE 'PROBLEM IS NOT IN THIS SUBROUT'
      RETURN
```

```
C        *********************************************
C        ***                                       ***
C        ***  SUBROUTINE EIGEN(A,R,N,NA,NR,MU)     ***
C        ***                                       ***
C        *********************************************
C
         SUBROUTINE EIGEN(A,R,N,NA,NR,MU)
C
C       THIS SUBROUTINE COMPUTES THE EIGEN VALUES
C        AND EIGEN VECTORS OF MATRIX A
C
C       DESCRIPTION OF PARAMETERS
C     A - ORIGINAL MATRIX(SYMMETRIC). DESTROYED IN COMPUTATION.
C         RESULTANT EIGEN VALUES ARE DEVELOPED IN DIAGONAL OF
C         MATRIX A IN DESCENDING ORDER.
C     R - RESULTANT MATRIX OF EIGENVECTORS (STORED COLUMNWISE,
C         IN SAME SEQUENCE AS EIGEN VALUES)
C       MU - INPUT CODE
C      0   COMPUTE EIGENVALUES AND EIGENVECTORS
C      1   COMPUTE EIGENVALUES ONLY, R NEED NOT BE DIMENSIONED
C      THE MAIN PROGRAM BUT MUST APPEAR IN THE CALLING SEQUENCE.
C       NA=N*(N+1)/2      SHOULD BE DEFINED IN MAIN PROGRAM
C       NR=N*N            SHOULD BE DEFINED IN MAIN PROGRAM
C
          DIMENSION A(NA),R(NR)
5         RANGE=1.0E-6
          IF(MU-1) 10,25,10
10        IQ=-N
          DO 20 J=1,N
          IQ=IQ+N
          DO 20 I=1,N
          IJ=IQ+I
          R(IJ)=0.0
          IF(I-J) 20,15,20
15        R(IJ)=1.0
20        CONTINUE
C
C       COMPUTE INITIAL AND FINAL NORMS (ANORM AND ANORMX)
C
25        ANORM=0.0
          DO 35 I=1,N
          DO 35 J=I,N
          IF(I-J) 30,35,30
30        IA=I+(J*J-J)/2
          ANORM=ANORM+A(IA)*A(IA)
35        CONTINUE
          IF(ANORM) 165,165,40
40        ANORM=1.414*SQRT(ANORM)
          ANRMX=ANORM*RANGE/FLOAT(N)
C
C         INITIALIZE INDICATORS AND COMPUTE THRESHOLD, THR
C
          IND=0
          THR=ANORM
45        THR=THR/FLOAT(N)
```

```
50        L=1
55        M=L+1
C
C         COMPUTE SIN AND COS
C
60        MQ=(M*M-M)/2
          LQ=(L*L-L)/2
          LM=L+MQ
62        IF(ABS(A(LM))-THR) 130,65,65
65        IND=1
          LL=L+LQ
          MM=M+MQ
          X=0.5*(A(LL)-A(MM))
68        Y=-A(LM)/SQRT(A(LM)*A(LM)+X*X)
          IF(X) 70,75,75
70        Y=-Y
75        SINX=Y/SQRT(2.0*(1.0+(SQRT(1.0-Y*Y))))
          SINX2=SINX*SINX
78        COSX=SQRT(1.0-SINX2)
          COSX2=COSX*COSX
          SINCS=SINX*COSX
C
C         ROTATE L AND M COLUMNS
C
          ILQ=N*(L-1)
          IMQ=N*(M-1)
          DO 125 I=1,N
          IQ=(I*I-I)/2
          IF(I-L) 80,115,80
80        IF(I-M) 85,115,90
85        IM=I+MQ
          GO TO 95
90        IM=M+IQ
95        IF(I-L) 100,105,105
100       IL=I+LQ
          GO TO 110
105       IL=L+IQ
110       X=A(IL)*COSX-A(IM)*SINX
          A(IM)=A(IL)*SINX+A(IM)*COSX
          A(IL)=X
115       IF(MV-1) 120,125,120
120       ILR=ILQ+I
          IMR=IMQ+I
          X=R(ILR)*COSX-R(IMR)*SINX
          R(IMR)=R(ILR)*SINX+R(IMR)*COSX
          R(ILR)=X
125       CONTINUE
          X=2.0*A(LM)*SINCS
          Y=A(LL)*COSX2+A(MM)*SINX2-X
          X=A(LL)*SINX2+A(MM)*COSX2+X
          A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)
          A(LL)=Y
          A(MM)=X
C
C         TESTS FOR COMPLETION
```

```
C
C       TEST FOR M = LAST COLUMN
C
130     IF(M-N) 135,140,135
135     M=M+1
        GO TO 60
C
C       TEST FOR L = SECOND FROM LAST COLUMN
C
140     IF(L-(N-1)) 145,150,145
145     L=L+1
        GO TO 55
150     IF(IND-1) 160,155,160
155     IND=0
        GO TO 50
C
C       COMPARE THRESHOLD WITH FINAL NORM
C
160     IF(THR-ANRMX) 165,165,45
C
C       SORT EIGENVALUES AND EIGENVECTORS
C
165     IQ=-N
        DO 185 I=1,N
        IQ=IQ+N
        LL=I+(I*I-I)/2
        JQ=N*(I-2)
        DO 185 J=I,N
        JQ=JQ+N
        MM=J+(J*J-J)/2
        IF(A(LL)-A(MM)) 170,185,185
170     X=A(LL)
        A(LL)=A(MM)
        A(MM)=X
        IF(MV-1) 175,185,175
175     DO 180 K=1,N
        ILR=IQ+K
        IMR=JQ+K
        X=R(ILR)
        R(ILR)=R(IMR)
180     R(IMR)=X
185     CONTINUE
        RETURN
```

```
C     *************************************************************
C     *                                                           *
C     *SUBROUTINE DIST(PHIREF,PHIX,ALAMD,NFEAT,NPTS,DRX)          *
C     *                                                           *
C     *************************************************************
C
C     THIS SUBROUTINE COMPUTES THE DISTANCE BETWEEN THE
C     REFERENCE AND TEST ORTHOGONAL PARAMETERS USING THE
C     ORTHOGONAL DISTANCE METRIC.
C
C     PHIREF IS THE REFERENCE ORTHOGONAL PARAMETER SET
C
C     PHIX IS THE TEST ORTHOGONAL PARAMETER SET
C
C     ALAMD IS THE REFERENCE EIGENVALUE SET
C
C     NFEAT IS THE NUMBER OF PARAMETERS USED IN THE
C     DISTANCE COMPUTATION
C
C     NPTS IS THE NUMBER OF POINTS PER PARAMETER (=1)
C
C     DRX IS THE ORTHOGONAL DISTANCE COMPUTED
C
      SUBROUTINE DIST(PHIREF,PHIX,ALAMD,NFEAT,NPTS,DRX)
      DIMENSION PHIREF(NFEAT,NPTS),PHIX(NFEAT,NPTS),ALAMD(9)
      DRX=0.0
      DO 1 I=1,NFEAT
      DO 2 J=1,NPTS
      DIFF=(PHIREF(I,J)-PHIX(I,J))
      DIFFSQ=(DIFF*DIFF)/ALAMD(I)
      DRX=DRX+DIFFSQ
2     CONTINUE
1     CONTINUE
      RETURN
```

```
C        *********************************************
C        ***                                       ***
C        ***  SUBROUTINE CODNAM(I,IPOS,IMARK)      ***
C        ***                                       ***
C        *********************************************
C
      THIS SUBROUTINE PRINTS THE NAMES OF THE
C      CODING SCHEMES USED FOR THE TESTING (ACTUAL) AND THE
C      NAMES OF THE CODING SCHEMES AS IDENTIFIED BY THE
C      CLASSIFICATION STAGE OF THE ALGORITHIM.
C
      SUBROUTINE CODNAM(I,IPOS,IMARK)
      IF(IMARK.NE.1) GO TO 500
      IF(I.NE.1) GO TO 200
      WRITE(10,131)
      WRITE(12,131)
      GO TO 210
 131  FORMAT(10X,'CODING SCHEME EMPLOYED IN TEST SIGNAL
     &:-PAM')
 200  IF(I.NE.2) GO TO 201
      WRITE(10,132)
      WRITE(12,132)
      GO TO 210
 132  FORMAT(10X,' CODING SCHEME EMPLOYED IN TEST SIGNAL
     @:- FIRST ORDER FIXED DPAM')
 201  IF(I.NE.3) GO TO 202
      WRITE(10,133)
      WRITE(12,133)
      GO TO 210
 133  FORMAT(10X,' CODING SCHEME EMPLOYED IN TEST SIGNAL
     #:-FIRST ORDER ADAPTIVE DPAM')
 202  IF(I.NE.4) GO TO 203
      WRITE(10,134)
      WRITE(12,134)
      GO TO 210
 134  FORMAT(10X,' CODING SCHEME EMPLOYED IN TEST SIGNAL
     $:- SECOND ORDER ADAPTIVE DPAM')
 203  WRITE(10,135)
      WRITE(12,135)
      GO TO 210
 135  FORMAT(5X,' CODING SCHEME EMPLOYED IN TEST SIGNAL
     %:- LOG PAM (U=100)')
 500  IF(IPOS.NE.1) GO TO 300
      WRITE(10,140)
      WRITE(12,140)
 140  FORMAT(5X,'TEST CODE IDENTIFIED AS :- PAM')
      GO TO 310
 300  IF(IPOS.NE.2) GO TO 301
      WRITE(10,141)
      WRITE(12,141)
 141  FORMAT(5X,'TEST CODE IDENTIFIED AS
     #:- FIRST ORDER FIXED DPAM')
      GO TO 310
 301  IF(IPOS.NE.3) GO TO 302
      WRITE(10,142)
```

```
      WRITE(12,142)
142   FORMAT(5X,'TEST CODE IDENTIFIED AS
     &:- FIRST ORDER ADAPTIVE DPAM')
      GO TO 310
302   IF(IPOS.NE.4) GO TO 303
      WRITE(10,143)
      WRITE(12,143)
143   FORMAT(5X,'TEST CODE IDENTIFIED AS
     &:- SECOND ORDER ADAPTIVE DPAM')
      GO TO 310
303   WRITE(10,144)
      WRITE(12,144)
144   FORMAT(5X,'TEST CODE IDENTIFIED AS
     $:- LOG PAM U=100')
210   CONTINUE
310   CONTINUE
      RETURN
```

```
C        *************************************************
C        ***                                           ***
C        ***   SUBROUTINE SCPLOT(X,Y,N,NOBSV)          ***
C        ***                                           ***
C        *************************************************
C
C        THIS SUBROUTINE IS USED FOR GENERATING THE
C        THE SCATTERPLOTS.
C        THESE PLOTS ARE OBTAINED BY PLOTTING THE
C        X ARRAY OF POINTS AGAINST THE Y ARRAY
C
C        X REPRESENTS THE ARRAY OF VALUES TO BE PLOTTED
C        ALONG THE X-AXIS
C
C        Y REPRESENTS THE ARRAY OF VALUES TO
C        BE PLOTTED ALONG THE Y-AXIS
C
C        N IS THE TOTAL NUMBER OF POINTS IN EACH ARRAY
C
C        NOBSV IS THE NUMBER OF POINTS IN EACH ARRAY
C        FOR WHICH THE SYMBOL WHICH IS USED IN THE
C        PLOT REMAINS THE SAME.
C        FOR EXAMPLE, IF INUM=N/NOBSV,
C        THEN THERE WILL BE INUM DIFFERENT
C        SYMBOLS BEING USED IN THE PLOT.
C        THE FIRST SYMBOL WILL BE USED FOR
C        THE FIRST NOBSV POINTS AND THE NEXT
C        SYMBOL WILL BE USED FOR THE NEXT
C        NOBSV POINTS IN THE ARRAYS.
C
      SUBROUTINE SCPLOT(X,Y,N,NOBSV)
      DIMENSION X(N),Y(N),NCODE(1200),GL(2)
      DIMENSION VL(2),DT(2),XP(11),RC(62)
      INTEGER ANG(20),BLK,A(150),NAME(130)
      DATA ANG,BLK/'1','2','3','4','5'
     $,'6','7','8','9','0','1','2'
     #  ,'3','4','5','6','7','8','9','0',' '/
      DATA RC/1.E29,1.E28,1.E27,1.E26,1.E25,
     @1.E24,1.E23,1.E22,
     #  1.E21,1.E20,1.E19,1.E18,1.E17,1.E16,
     @1.E15,1.E14,1.E13,1.E12,
     #  1.E11,1.E10,1.E9,1.E8,1.E7,1.E6,
     @1.E5,1.E4,1.E3,1.E2,1.E1,1.E0,
     #  1.E-1,1.E-2,1.E-3,1.E-4,1.E-5,
     @1.E-6,1.E-7,1.E-8,1.E-9,1.E-10,
     #  1.E-11,1.E-12,1.E-13,1.E-14,
     @1.E-15,1.E-16,1.E-17,1.E-18,1.E-19,
     #  1.E-20,1.E-21,1.E-22,1.E-23,1.E-24,
     @1.E-25,1.E-26,1.E-27,1.E-28,
     #  1.E-29,1.E-30,1.E-31,1.E-32/
      XMIN=X(1)
      XMAX=X(1)
      YMIN=Y(1)
      YMAX=Y(1)
      NSYM=1
```

```
      NCODE(1)=ANG(1)
      DO 3 I=2,N
      RATIO=FLOAT(I)/FLOAT(NOBSV)
      IRATIO=IFIX(RATIO)
      DIFF=RATIO-IRATIO
      IF(DIFF.EQ.0) GO TO 83
      NCODE(I)=ANG(NSYM)
      GO TO 36
83    NCODE(I)=ANG(NSYM)
      NSYM=NSYM+1
36    XMAX=AMAX1(XMAX,X(I))
      XMIN=AMIN1(XMIN,X(I))
      YMAX=AMAX1(YMAX,Y(I))
      YMIN=AMIN1(YMIN,Y(I))
3     CONTINUE
      GL(1)=XMAX-XMIN
      GL(2)=YMAX-YMIN
      DO 10 J=1,2
      DT(J)=FLOAT(J)*.01*GL(J)
      IF(DT(J).GT.1.E30) GO TO 8
      DO 5 I=1,62
      IF(DT(J).GT.RC(I)) GO TO 6
5     CONTINUE
      DT(J)=1.E-32
      GO TO 10
6     CR=RC(I)
      DO 7 K=2,8
      IF(K.EQ.7) GO TO 7
      IF(DT(J).LT.FLOAT(K)*CR) GO TO 9
7     CONTINUE
      DT(J)=RC(I-1)
      GO TO 10
8     WRITE(12,100)
      RETURN
9     DT(J)=FLOAT(K)*CR
10    VL(J)=.5*(FLOAT(3-J)*50.*DT(J)-GL(J))
      BOTX=XMIN-VL(1)
      TOPY=YMAX+VL(2)
      U=ABS(BOTX)
      REM=AMOD(U,DT(1))
      U=U-REM
      BOTX=SIGN(U,BOTX)
      U=ABS(TOPY)
      REM=AMOD(U,DT(2))
      U=U-REM
      TOPY=SIGN(U,TOPY)+DT(2)
      DXP=10.*DT(1)
      DO 17 I=1,11
17    XP(I)=BOTX+FLOAT(I-1)*DXP
      WRITE(12,101)
      DY2=.51*DT(2)
      DO 22 JJ=1,51
      I=JJ-1
      DO 165 J=1,101
165   A(J)=BLK
```

```
      YN=TOPY-FLOAT(I)*DT(2)
      DO 18 J=1,N
      IF(ABS(YN-Y(J)).GT.DY2) GO TO 18
      L=1+IFIX(.5+(X(J)-BOTX)/DT(1))
      A(L)=NCODE(J)
      GO TO 18
18    CONTINUE
      IF(MOD(I,5).EQ.0) GO TO 20
      WRITE(12,102) (A(J),J=1,101)
      GO TO 22
20    WRITE(12,103) YN,(A(J),J=1,101)
22    CONTINUE
      WRITE(12,104) (XP(I),I=1,11)
      WRITE(12,105) XMAX,XMIN,YMAX,YMIN,DT(1),DT(2)
100   FORMAT(1H1,10X,20HPLOT RANGE TOO LARGE)
101   FORMAT(1H1,12X,1H*,10(9X,1H*)/12X,103(1H*))
102   FORMAT(12X,1H*,101A1,1H*)
103   FORMAT(1X,E9.2,3H **,101A1,2H**)
104   FORMAT(12X,103(1H*)/4X,11(9X,1H*)/8X,11(1X,E9.2))
105   FORMAT(1X,6HXMAX =,E12.5,2X,6HXMIN =
     #,E12.5,2X,6HYMAX =,E12.5,2X,
     @  6HYMIN =,E12.5,2X,7HXINCR =,E12.5,2X
     $,7HYINCR =,E12.5///)
      RETURN
```

REFERENCES

1. N.S. JAYANT 'Digital Coding of Speech Waveforms : PCM, DPCM and DM Quantizers, Proc. IEEE,Vol.62, pp.611-632, May 1974

2. L.R. RABINER & R.W. SCHAFER, 'Digital Processing of Speech Signals', McGraw-Hill Book Publishing Company, 1976

3. HAYKIN 'Communication Systems' Prentice Hall Inc., Englewood Cliffs, N.J., 1982

4. MARTIN S. RODEN, 'Digital and Data Communication Systems' Prentice Hall, Inc. Englewood Cliffs, N.J., 1982

5. ALAN V. OPPENHEIM, 'Applications of Digital Signal Processing' Prentice-Hall, Inc., Englewood Clifts, N.J., 1978

6. GTE LENKURT ELECTRIC, 'Pulse Code Modulation', 1973

7. J.S. BENDAT & A.G.PIERSOL, 'Random Data : Analysis and Measurement' Wiley-Interscience, 1971

8. DUDA & HART, 'Pattern Classification and Scene Analysis' Wiley-Interscience, 1973

9. WATANABE, S., 'Karhunen Loeve Expansion and Factor Analysis: Theoretical Remarks and Applications' Trans. Fourth Prague Conf. Inform. Theory, Statistical Decision Functions and Random Processes, 635-660 (1965)

10. BISHNU S. ATAL, 'Automatic Recognition of Speakers from their Voices' Proc. of IEEE, Vol.64, pp. 460-475, April 1976

11. MARVIN S. SAMBUR, 'Speaker Recognition Using Orthogonal Linear Prediction' IEEE Trans. ASSP-24, pp.283-289 Aug. 1976

VITA AUCTORIS

VIJAY VENKATACHALAM

1960	Born on 10 September in Bangalore, India
1975	Passed the Indian School Certificate Examination (affiliated to the University of Cambridge, U.K.) from the Frank Anthony Public School, Bangalore, India
1981	Graduated with Bachelor of Engineering (B.E.) in Electronics & Communication Engg. from the Regional Engineering College, Tiruchirapalli, India
1982	Project Assistant at the School of Automation, Indian Institute of Science, Bangalore, India
1984	Candidate for the degree of Master of Applied Science in Electrical Engineering at the University of Windsor, Ontario, Canada